THE UNIVERSITY OF NEWCASTLE UPON TYNE
DEPARTMENT OF COMPUTING SCIENCE

UNIVERSITY OF
NEWCASTLE UPON TYNE

Performance and Reliability Modelling of Computing Systems Using Spectral Expansion

by

Ram Chakka

PhD Thesis

September 1995

Declaration:

I certify that no part of the original material offered here, in this thesis, has previously been submitted by me for a degree or any other qualification, in this or any other University.

RAM CHAKKA

Dedicated to, Sri Sathya Sai Baba

Abstract

This thesis is concerned with the analytical modelling of computing and other discrete event systems, for steady state performance and dependability. That is carried out using a novel solution technique, known as the *spectral expansion method*. The type of problems considered, and the systems analysed, are represented by certain two-dimensional Markov-processes on finite or semi-infinite lattice strips. A sub set of these Markov processes are the *Quasi-Birth-and-Death* processes.

These models are important because they have wide ranging applications in the design and analysis of modern communications, advanced computing systems, flexible manufacturing systems and in dependability modelling. Though the matrix-geometric method is the presently most popular method, in this area, it suffers from certain drawbacks, as illustrated in one of the chapters. Spectral expansion clearly rises above those limitations. This also, is shown with the aid of examples.

The contributions of this thesis can be divided into two categories. They are,

- The theoretical foundation of the spectral expansion method is laid. Stability analysis of these Markov processes is carried out. Efficient numerical solution algorithms are developed. A comparative study is performed to show that the spectral expansion algorithm has an edge over the matrix-geometric method, in computational efficiency, accuracy and ease of use.

- The method is applied to several non-trivial and complicated modelling problems, occuring in computer and communication systems. Performance measures are evaluated and optimisation issues are addressed.

Key-Words: *Spectral Expansion, Markov Processes, Multi-processors, Performance, Reliability, Dependability, Networks of Queues,*

Acknowledgements

I gratefully thank Professor Isi Mitrani for introducing me to this interesting area of research and for his outstanding supervision, throughout this thesis. The plentiful technical discussions I had with him, have been highly stimulating, very productive, and have given me a great deal of professional satisfaction. Isi has been considerate, and ever ready to discuss.

The suggestions and challenges posed by some of the anonymous referees of our papers have considerably benefitted this research, for we could meet all those challenges. I am thankful to those referees. Some remarks of Professor P.J. Courtois, on the early part of this work, have also been helpful.

While I was busy modelling breakdowns and repairs, whenever my computer system failed Stuart Wheater has come up graciously, and restored the system. I thank him for this. I thank Robert Burnett for several useful suggestions on LaTeX. Shirley Craig, the departmental librarian, has been helpful.

This research work was carried out as part of the deliverables for the ESPRIT Basic Research projects PDCS and PDCS2, in which I was employed as a Research Associate for over half a decade.

My parents and brothers have been supportive. My friend, B.B. Prahlada Rao, in India, has been supportive during all these years.

During difficult times, I received the necessary Grace from the Almighty, strengthening my Faith in God.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Performance and reliability modelling

Computers have become very important not only in industry and business but also in everyday living. When we draw cash from a cash point we use computers, travel reservation systems and air traffic controls use computers, hospitals have them in diagnostic equipment and in patient monitoring systems. At present, there is a growing emphasis not only on the design aspects but also on how to use them effectively and efficiently. Reliability is clearly a measure of effectiveness. Fault tolerant computing systems are designed for greater reliability and dependability. Ultra-high reliability requirements are demanded in critical applications such as aircraft control and nuclear plant control systems.

Computer reliability is usually achieved by hardware or component reliability, hardware redundancy, using repairable systems and reliable architectures. On the software side, producing reliable software and employing software fault tolerance are the key to achieve reliable computation. Multiprocessor systems are more reliable over single processor ones because processor failures(not all of them) do not result in system failure but only in a degraded mode of operation. Multiprocessors are very important for parallel processing applications also.

A fault tolerant computing system is not only to be designed, but needs to be evaluated for its performance. Reliability, speed of computation, throughput rate, and response time distributions are some of the measures that are usually sought. Performance modelling is essentially probablistic and mathematical modelling. There are several different as well as complementary techniques of this, namely bench marking, measurement based analysis and extrapolation [3, 29, 61], analytical modelling employing queueing theory and Markov chain analysis [20], simulation of the system's working by a computer program and also a recent and highly promising method based on perturbation analysis on simulated or real data [27]. In highly reliable systems the failures of components are rare events. In such systems naive simulation is inefficient and at times even impossible due to the rarity of system failure events. Importance sampling is used to simulate such systems approximately [58, 23]. Also, contributions exist to simulate discrete event systems efficiently using parallel algorithms on parallel processors [24].

The present thesis work falls in the category of analytical modelling. Analytical methods employ certain simplified assumptions of the system to achieve mathematical tractability. In queueing systems, these include assumptions regarding the probablistic distributions of job arrivals, service times, failure and repair characteristics. Past experience has shown that analytical queueing models are extremely effective and accurate enough to analyse or predict system performance, in spite of these not so realistic assumptions [59]. Techniques such as probablistic analysis, representing the behaviour of the system by finite or infinite Markov or semi-Markov processes, use of queueing theoretic models, optimization techniques form the basis for analytical modelling. The advantage of analytical modeling over other techniques is their computational efficiency. Quite often, simulation and other techniques need a great deal of computation and other resources that make the optimal system design difficult or even impossible. The flexibility and applicability of analytical modelling is somewhat limited.

1.2 Systems of interest

The systems that are being considered for a detailed study and research in this thesis can be modelled by a class of Markov processes. These are two dimensional Markov processes on finite or semi-infinite strips. Several practical examples can be thought of in this frame work. Consider as our first example, a multiprocessor system serving a stream of incoming jobs. The multiprocessor consists of N homogeneous processors that are prone to independent failures. A processor, when fails, is immediately set into repair, and once repaired the processor is back in service. Thus, at any given time the multiprocessor is *operative* with a number of processors ranging from 0 to N. We can call these $N + 1$ states, the *operative* states of the multiprocessor. There is an unbounded queue which the incoming jobs join. Processed jobs leave the system. Jobs are homogeneous and served one at a time by a processor. If a job is pre-empted due to the processor failure, its service starts whenever a processor is available. In this case the service policy is either *resume* or *repeat with re-sampling*. The arrivals of jobs is governed by a Poisson process and the processor service, failures, repairs are exponential. In such a system, it can be easily seen that the state of the system at any given time can be sufficiently described by two integer-valued random variables. The first of them is a finite one, ranging from 0 to N, indicating the operative state of the multiprocessor. The second random variable represents the number of jobs present in the system, and ranges from 0 to ∞. Thus, all the states of this system form a Markov process on a semi-infinite rectangular lattice strip with a width of $N + 1$.

Another illustration of the type is possible through the example of a serial transfer line with two machines, M1 and M2, and a finite buffer, B, between them [35]. Each job is to be processed by M1 and then by M2, in order. Jobs to be processed arrive at M1 and form an unbounded queue. After being processed by M1 they are placed in the buffer B in order to wait to be processed by M2 when the latter is available. The size of B is finite, hence when B is full M1 stops working. Hence, M1 is *blocked* due to finite buffer. Processed jobs from M2 leave the system. Job arrivals to M1 are assumed Poisson, service times exponential and service discipline FCFS. Clearly, the state of this system at any given time can be represented by just two integer-valued random variables. The first of them, a finite one, is the number of jobs held in the buffer including the one, if any, currently processed by M2. The second random variable is the number of jobs in the queue before M1 including the job held by M1. If the size of the buffer B is N, including the one held by M2, then the first random variable ranges from 0 to N. The second random variable has the range from 0 to ∞ . Thus, here again we get a Markov process on a semi-infinite lattice strip of width $N + 1$. The possible transitions from a given state into different other states can be worked out easily by studying the system's working. If the queue before M1 is bounded, then the Markov process would be on a finite rectangular strip. Some of the performance measures that can be sought are, the distribution of queue lengths before each of the machines, utilisations of the machines, mean waiting time and total turn-around time of jobs. If the Markov process can be analysed for the state probabilities, then the required performance measures can be computed using the state probabilities. This Markov model can be extended to transfer lines with more than two machines, but with an increase in the computational complexity. For example, in 3 machine tranfer line, if $N1$ and $N2$ are the sizes of the two buffers respectively, then the resulting Markov process will be on a semi-infinite strip of width $(N1 + 1) * (N2 + 1)$.

Perhaps yet another example is worth illustrating in order to see the wide range of problems captured by these Markov processes [14]. As the third and the last example, we take up the case of a machine shop. This machine shop has two machines, M1 and M2, that are different from each other. Jobs arrive at the machine shop according to a Poisson distribution. Waiting jobs form an unbounded queue. Jobs are homogeneous and are served in FCFS order. Each job consists of two tasks, namely Task 1 and Task 2. Task 1 has to be performed first and then only Task 2 is to be performed. Once the Task 2 of a job is completed, it leaves the system as a finished job. M1 performs only Task 1 and M2 can perform either of the tasks. M2 is faster than M1 in performing Task 1. After Task 1 is completed, a job is to be scheduled for Task 2 on M2 when the latter is available. Thus, at any given time M1 can be either idle or performing Task 1 on a job, and, M2 can be idle or performing Task 1 on a job or performing Task 2 on a job whose Task 1 is already completed. Several alternate strategies can be thought of regarding the scheduling of jobs on M1 and M2. We choose the following strategy:

(a) When the Task 1 of a job is completed on M2, then the job is automatically scheduled for Task 2 on M2.

(b) When the Task 1 of a job, say J1, is completed on M1, then its Task 2 is scheduled immediately if M2 is either available or performing Task 1 of a different job, J2. In the latter case, J2 is scheduled on M1 with either *resume* or *repeat with re-sampling* service policy.

(c) When the Task 1 of a job, J1, is completed on M1, if M2 performing Task 2 of a different job, then J1 is retained by M1 till M2 is available. Here M1 is idle but retains J1. When M2 becomes available, J1 is scheduled on M2 and M1 goes for Task 1 of a new job if available.

(d) When there is only one job in system with Task 1 to be performed, it is scheduled on M2 only since it is faster than M1.

Clearly there are six performing or *operative* states of the machine shop. They are, (Ml-idle, M2-idle), (Ml-idle, M2-Task 1), (Ml-Task 1, M2-Task 1), (Ml-idle, M2-Task 2), (Ml-idle & retaining a job, M2-Task 2), (Ml-Task 1, M2-Task 2). The state of the system can then be represented by two integer valued random variables, one of them representing the operative state of the machine shop and the other the number of jobs in the system. The second random variable varies from 0 to ∞. Once again we get a Markov process on a semi-infinite strip, with width 6, that models the system in consideration. If the queue is bounded, then the Markov process would be on a finite rectangular strip.

1.3 Literature survey

The type of systems described above are all modelled by two-dimensional Markov processes on semi-infinite or finite strips. In the case of the former, let I, J be the finite and infinite integer valued random variables respectively. These processes occur often in modelling computing systems, communication systems and several other discrete event systems. Significant research work has been carried out in this area [52] and there is a sufficient need for further work.

Our primary interest is a sub-class that has the following properties:

(a) the instantaneous transition rates out of state (i, j) do not depend on j when the latter is sufficiently large, say for $j > M$.

(b) the jumps of the random variable J are of limited size.

There are many sub-classes of this that have widely appeared in the literature. When the jumps of the random variable J, in a transition, are either 1 or 0 or -1, these processes are known as *Quasi-Birth-and-Death* type [66]. In the QBD process, if the non-zero jumps in J are not accompanied with changes in I, then these processes are known as *Markov-modulated Birth and Death* processes [53]. The infinite number of states involved makes the solution of these models non-trivial.

There are several methods of solving these models, either the whole class of models or any of the sub-classes. Seelen has analysed a Ph/Ph/c queue in this frame work [55]. Seelen's method is an approximate one, the Markov chain is first truncated to a finite state which is an approximation of the original process. The resulting finite state Markov chain is then analysed, by exploiting the structure in devising an efficient iterative solution algorithm.

The second method is to reduce the infinite-state problem to a linear equation involving vector generating function and some unknown probabilities. The latter are then determined with the aid of the singularities of the coefficient matrix. A comprehensive treatment of that approach, in the context of a discrete-time process with a general M/G/1 type structure, is presented in [17].

The third method, which is the main technique that is applied throughout this thesis, is known as *spectral expansion* method. It is based on expressing the invariant vector of the process in terms of eigenvalues and left eigenvectors of a certain matrix polynomial. Some of the ideas concerning spectral expansion method have been known for some time [51, 60], but there have not been many examples in the performance literature. Takacs's [60] works concentrate mostly on one-dimensional Markov processes, with one random variable. Neuts [51] does deal with two-dimensional ones, yet has not given efficient computational algorithms for spectral expansion. Some recent works on it have been reported in [13, 45]. An efficient algorithmic implementation for probability distribution is reported in [45, 5]. Earliest numerical results using this solution algorithm are in [5, 7]. The generating function and the spectral expansion methods are closely related. However, the latter produces steady-state probabilities directly using an algebraic expansion while the former provides them through a transform.

The fourth way of solving these models is the well known *matrix-geometric* method, first proposed by Evans [15, 66, 51]. In this method a non-linear matrix equation is first formed from the system parameters and the minimal nonnegative solution R of this equation is computed by an iterative method. The invariant vector is then expressed in terms of the powers of R. Neuts claims this method has probablistic interpretation for the steps in computation. That is certainly an advantage. Yet, this method suffers from the fact that there is no way of knowing how many iterations are needed to compute R to a given accuracy. It can also be shown that for certain parameter values the computation requirements are uncertain and formidably large.

1.4 Organisation of the thesis

This thesis is concerned with the performance evaluation of computer and other discrete event systems that can be modelled by two dimensional Markov processes on semi-infinite or finite lattice strips. The contributions of this thesis are two-fold. Firstly, we develop and present the spectral expansion method and an efficient algorithmic implementation for the performance analysis of the type systems and the Markov processes, described in the preceding section. In this framework, a number of important and non-trivial problems are modelled and detailed analysis is made. In all these problems, spectral expansion is extensively made use of, apart from using other operations research techniques. Secondly, we make a systematic comparison of the effectiveness and applicability of the spectral expansion method, with that of the presently most popular method, the matrix-geometric method. This comparative study would be quite useful for modellers.

The organisation of the thesis is as follows. Chapter 2 describes in a good detail, the development of the spectral expansion method for infinite Markov processes, stability and spectral analysis, efficient computational algorithms, some aspects of implementation and also certain generalisations. Chapter 3 presents two non-trivial examples in order to illustrate the application of spectral expansion. An important extension of the method to solve Markov processes of large, *finite* state space, is developed in Chapter 4, and that is explained along with an example. In Chapter 5, a comparative study of the effectiveness of spectral expansion and the matrix-geometric method is carried out. In Chapter 6, a heterogeneous multiprocessor with breakdowns is analysed for performance measures, using spectral expansion. The importance of the repair strategy is illustrated and certain nearly optimal strategies are conceived. A performance study of these strategies is also done. The modelling and analysis of queues with breakdowns and bursty arrivals is taken up in Chapter 7. The bursty arrivals are represented by Markov-modulated arrival processes. Steady state analysis is made using the spectral expansion method. Also, a simple and and computationally effective model is suggested for the departure process of such a queue. The subject of Chapter 8 is a complex modelling problem: that of open queueing networks with breakdowns and repairs. There has not been any existing exact solution or acceptable approximate solution to this problem. Several approximate numerical solution techniques are developed for this problem, achieving considerable accuracy. In these solutions, the spectral expansion method is extensively used. Chapter 9 presents conclusions and directions for further research work, in this and related areas.

1.4.1 Publications

Many of the original contributions of thesis are either published already, or accepted for publication. Some of these papers are given in the Bibliography. They are, [4, 5, 6, 7, 8, 9, 10, 47].

Chapter 2

Spectral Expansion

2.1 Introduction

Spectral expansion is an emerging solution technique useful in performance and dependability modelling of discrete event systems. It solves Markov models of certain kind that arise in several practical system models. The reported applications and results include, performability modelling of several types of multiprocessors [5, 7], multi-task execution models [14], networks of queues with unreliable servers [10] and many other practical systems [13]. Though some preliminary ideas were known earlier [51], an efficient solution algorithm for spectral expansion was developed in [45, 5]. The earliest numerical results on this algorithm are reported in [5, 7]. This algorithm seems to fare better than the matrix-goemetric method in accuracy, speed and ease of use [47, 10]. In this chapter, we develop the mathematics of the spectral expansion method and present the solution algorithm.

2.2 The Markov model

The systems that are of interest are modelled by two-dimensional Markov processes on semi-infinite lattice strips. The state of the system at time t is denoted by two integer valued random variables, $I(t)$, $J(t)$; $I(t)$ taking a *finite* set of values and $J(t)$, an *infinite* set of values. Without loss of generality, we can assume the minimum value of $I(t)$ is 0 and N, the maximum. The minimum value of the random variable $J(t)$ is 0 and it can take values from 0 to ∞. The Markov process is denoted by $X = \{[I(t), J(t)] \; ; \; t \geq 0\}$. We assume this is irreducible with a state space $\{0, 1, \ldots, N\} \times \{0, 1, \ldots\}$. For a convenient representation, it is assumed that $I(t)$ varies in the lateral or horizontal direction and $J(t)$ is represented in the vertical direction of the semi-infinite rectangular lattice strip. The possible transitions that underlie this Markov process are given by :

(a) A_j – purely lateral transitions – from state (i, j) to state (k, j), $(0 \leq i, k \leq N \; ; \; i \neq k \; ; \; j = 0, 1, \ldots)$;

(b) B_j – one-step upward transitions – from state (i, j) to state $(k, j + 1)$, $(0 \leq i, k \leq N \; ; \; j = 0, 1, \ldots)$;

(c) C_j – one-step downward transitions – from state (i, j) to state $(k, j - 1)$, $(0 \leq i, k \leq N \; ; \; j = 1, 2, \ldots)$;

As it is seen above, the possible change in J in any transition is either $+1$ or 0 or -1. Later in this chapter, we shall also consider the Markov process Y in which multi-step jumps in J are possible. A_j, B_j and C_j are the transition rate matrices, square matrices each of size $(N + 1) \times (N + 1)$, associated with (a), (b) and (c) respectively. Thus, $A_j(i, k)$, $(i \neq k)$ is the transition rate from state (i, j) to state (k, j), and $A_j(i, i) = 0$. $B_j(i, k)$ is the transition rate from (i, j) to $(k, j + 1)$. $C_j(i, k)$ is the transition rate from (i, j) to $(k, j - 1)$, and $C_0 = 0$, by definition.

We assume the process has a threshold, an integer M, $(M \geq 1)$ such that the instantaneous transition rates of (a), (b) and (c) do not depend on j when $j \geq M$ in the case of (a), $j \geq M-1$ in the case of (b), and when $j \geq M$ for (c). In the examples given in the following chapters, it can be seen that such a threshold *does* exist in a large variety of real world problems occuring in computing and communication systems. Thus, we have

$$A_j = A,\ j \geq M\ ;\ B_j = B,\ j \geq M-1\ ;\ C_j = C,\ j \geq M\ ; \tag{2.1}$$

The states spanned by the finite r.v. $I(t)$ can be conveniently termed as the *operative states* of the system in consideration. The transitions in (a) are purely among the operative states, without any change in J, whereas the transitions in (b) and (c) are also among the operative states but with a fixed change in J by $+1$ and -1 respectively.

When the process is irredicible and the corresponding balance equations of the state probabilities have a unique normalizeable solution, we say the process is *ergodic* and there exists a steady state for that process. The objective of this analysis is to determine the steady state probability $p_{i,j}$ of the state (i,j) in terms of the known parameters of the system. $p_{i,j}$ is defined as:

$$p_{i,j} = \lim_{t\to\infty} P(I(t)=i, J(t)=j)\ ;\ i=0,1,\ldots,N\ ;\ j=0,1,\ldots \tag{2.2}$$

Let D_j^A, D_j^B and D_j^C be the diagonal matrices, of size $(N+1)\times(N+1)$ each, defined by their i^{th} diagonal element as,

$$D_j^A(i,i) = \sum_{k=0}^{N} A_j(i,k);\quad D_j^B(i,i) = \sum_{k=0}^{N} B_j(i,k);\quad D_j^C(i,i) = \sum_{k=0}^{N} C_j(i,k); \tag{2.3}$$

In other words, the i^{th} diagonal element of each of these diagonal matrices is the i^{th} row sum of the corresponding transition rate matrix. Then we also get similar diagonal matrices D^A, D^B and D^C for A, B and C respectively.

$$D^A(i,i) = \sum_{k=0}^{N} A(i,k);\quad D^B(i,i) = \sum_{k=0}^{N} B(i,k);\quad D^C(i,i) = \sum_{k=0}^{N} C(i,k); \tag{2.4}$$

2.3 The solution

All the states in a row of the lattice Markov process have the same value j for the unbounded r.v. $J(t)$. Similarly, any column consists of states with same i. Here, it is mathematically convenient to define the row vectors $\mathbf{v}_j$ as,

$$\mathbf{v}_j = (p_{0,j},\ p_{1,j}, \ldots,\ p_{N,j})\ ;\ j = 0, 1, \ldots \tag{2.5}$$

Thus, the elements of $\mathbf{v}_j$ are the probabilities of all states in a row, where $J = j$. In order to solve for the probability distribution $\{p_{i,j}\}$, it is necessary to solve the balance equations. It is mathematically more elegant to work with vectors $\mathbf{v}_j$ compared to $p_{i,j}$. The steady state balance equations satisfied by the vectors $\mathbf{v}_j$ are:

$$\mathbf{v}_j[D_j^A + D_j^B + D_j^C] = \mathbf{v}_{j-1}B_{j-1} + \mathbf{v}_jA_j + \mathbf{v}_{j+1}C_{j+1}\ ;\ j = 0, 1, \ldots, M-1 \tag{2.6}$$

$\mathbf{v}_{-1} = 0$ by definition. And, for $j \geq M$

$$\mathbf{v}_j[D^A + D^B + D^C] = \mathbf{v}_{j-1}B + \mathbf{v}_jA + \mathbf{v}_{j+1}C\ ;\ j = M, M+1, \ldots \tag{2.7}$$

These equations (2.7) are infinite in number. In addition to them, we have another equation resulting from the fact that all the probabilities $p_{i,j}$ sum to 1.0:

$$\sum_{j=0}^{\infty} \mathbf{v}_j\mathbf{e} = 1.0, \tag{2.8}$$

where $\mathbf{e}$ is the column vector of $N+1$ elements each of which is equal to 1. This definition of $\mathbf{e}$ is valid throughout this thesis.

It is the set of equations (2.7) that are infinite in number. Essentially, spectral expansion is the solution technique to solve these equations. The difference between the set (2.6) and the set (2.7) is that the former has coefficient matrices that are j-dependent whereas in the case of the latter the coefficient matrices are j-independent. The latter can be rewritten as

$$\mathbf{v}_jQ_0 + \mathbf{v}_{j+1}Q_1 + \mathbf{v}_{j+2}Q_2 = 0\ ;\ j = M-1, M, \ldots \tag{2.9}$$

where $Q_0 = B$, $Q_1 = A - D^A - D^B - D^C$ and $Q_2 = C$. This is a homogeneous vector difference equation of order 2, with constant coefficients. $Q(\lambda)$ is the *characteristic matrix polynomial* associated with this difference equation.

$$Q(\lambda) = Q_0 + Q_1\lambda + Q_2\lambda^2 \ . \qquad (2.10)$$

We also refer to $Q(\lambda)$ as the characteristic matrix polynomial of the Markov process X. The solution of (2.9) is closely related to the eigenvalues and the left-eigenvectors of $Q(\lambda)$. Henceforth, throughout this thesis, an eigenvalue means a finite eigenvalue and an eigenvector means a non-zero finite left-eigenvector. Let (λ, ψ) be an eigenvalue-eigenvector pair of $Q(\lambda)$, thus satisfying the equation:

$$\psi Q(\lambda) = 0 \ ; \ det[Q(\lambda)] = 0 \ . \qquad (2.11)$$

λ can be real or complex. If λ is real then ψ also is real, if λ is complex then ψ is complex. This is because the matrices Q_0, Q_1 and Q_2 are real. Also, if (λ, ψ) is a complex eigenvalue-eigenvector pair, then its conjugate pair (λ^*, ψ^*) also satisfies (2.11). $Q(\lambda)$ may also possess either a null eigenvalue or a multiplicity of null eigenvalues. Null eigenvalues are also refered as zero-eigenvalues throughout this thesis. If Q_0 is non-singular then $Q(\lambda)$ will have no zero-eigenvalues. However, if Q_0 is singular with a rank $N+1-r_0$ $(1 \leq r_0 \leq N+1)$ then it will have zero-eigenvalues with a multiplicity of r_0 and corresponding non-zero independent eigenvectors, if the sum of all principal minors of order $N+1-r_0$ of Q_0 is non-zero [37]. That condition is usually the case, hence, we assume the same. We also assume if Q_2 has a rank $N+1-r_2$, then the sum of all the principal minors of Q_2 of order $N+1-r_2$ is non-zero.

For a non-zero eigenvalue-eigenvector pair, (λ_k, ψ_k), by substituting $\mathrm{v}_j = \psi_k\lambda_k^j$ $(j \geq M-1)$ in the equations (2.7, 2.9), it can be easily seen that this set of infinite number of equations is satisfied. Hence, that is a *particular* solution. The expression may even be $\psi_k\lambda_k^{j+l_k}$ for any real l_k. These two expressions are equivalent since ψ_k

can always be scaled without affecting its property of being an eigenvector. If zero-eigenvalues do exist, let them be r_0 in number denoted by $\lambda_{z,k}$, $(k = 0, 1, \ldots, r_0 - 1)$ and the corresponding eigenvectors by $\psi_{z,k}$, $(k = 0, 1, \ldots, r_0 - 1)$. Then clearly, for every k, $(k = 0, 1, \ldots, r_0 - 1)$, $\mathrm{v}_j = \psi_{z,k}\lambda_{z,k}^{j-M+1}$, $(j \geq M - 1)$ satisfies (2.7, 2.9). It is assumed $0^0 = 1$. That expression can also be written as $\mathrm{v}_j = 0$, if $j \neq M - 1$ and $\psi_{z,k}$, if $j = M - 1$. This also is a *particular* solution. Yet, what is necessary is the general solution of these equations that also satisfy (2.6, 2.8).

Let us assume that $Q(\lambda)$ has d pairs of eigenvalue-eigenvectors. These can be real or in pairs of complex-conjugates. There can be simple or multiple eigenvalues. However, an assumption is made that when an eigenvalue is of multiplicity n, it does have n independent eigenvectors. In all our applications and experiments with several system models, this is what has been observed. Let (λ_k, ψ_k) be the k^{th} eigenvalue-eigenvector pair, where k varies from 0 to $d - 1$. Then it is easy to see the general solution for v_j of the equations (2.7, 2.9) can be a linear sum of all the factors $(\psi_k\lambda_k^{j-M+1})$. In other words,

$$\mathrm{v}_j = \sum_{k=0}^{d-1} a_k\psi_k\lambda_k^{j-M+1} \; ; \; j = M - 1, M, \ldots \tag{2.12}$$

In the state-probability form:

$$p_{i,j} = \sum_{k=0}^{d-1} a_k\psi_k(i)\lambda_k^{j-M+1} \; ; \; j = M - 1, M, \ldots \tag{2.13}$$

where a_k $(k = 0, 1, \ldots, d - 1)$ are *arbitrary* constants, some of them can be complex too.

In order to get the relevant solution from the general solution, let us consider some of the known steady state properties of the probability distribution. We know the sum of the probabilities of all states in any column is less than 1 though these states are infinitely many. Now consider the probability sum,

$$\sum_{j=M-1}^{\infty} p_{i,j} = \sum_{j=M-1}^{\infty} \sum_{k=0}^{d-1} a_k\psi_k(i)\lambda_k^{j-M+1}$$

The above is the sum of probabilities of all states in the i^{th} column that are above $j \geq M-1$. It can be easily seen that the necessary condition that this sum is *bounded* is

$$a_k = 0, \ if \mid \lambda_k \mid \geq 1.$$

With this, by suitably numbering the eigenvalues, the general solution is modified as,

$$\mathbf{v}_j = \sum_{k=0}^{c-1} a_k \psi_k \lambda_k^{j-M+1} \ ; \ j = M-1, M, \ldots \qquad (2.14)$$

This is equivalent to:

$$p_{i,j} = \sum_{k=0}^{c-1} a_k \psi_k(i) \lambda_k^{j-M+1} \ ; \ j = M-1, M, \ldots \qquad (2.15)$$

where c is the number of eigenvalues that are present *strictly within* (also termed as within, strictly inside or inside) the unit circle. These eigenvalues appear some as real and others as complex-conjugate pairs, and as do the corresponding eigenvectors. When the eigenvalues are complex-conjugate the corresponding constants also are complex-conjugate. This is so because, then only the right-hand side of equations (2.14, 2.15) would be real. We also have another condition to be satisfied always. That condition is the right-hand sides of equations (2.14, 2.15) are also positive. This can happen if the real parts of λ_k, ψ_k and a_k are positive. Indeed, this is what has been observed in all our experiments.

Now it remains to determine the constants a_k that are still unknown, to complete the solution. Their number c also is not known so far. Yet, we can now find $\mathbf{v}_{M-1}$, $\mathbf{v}_M$, ... as linear expressions in the c unknowns, a_k, $(k = 0, 1, \ldots, c-1)$. The other unknowns are, $\mathbf{v}_0$, $\mathbf{v}_1, \ldots, \mathbf{v}_{M-2}$. To get these unknowns, we have equations (2.6) for $j = 0, 1, \ldots, M-1$. This is a set of $M \times (N+1)$ linear equations with $(M-1) \times (N+1)$ unknown probabilities (all the probabilities in v_j, $(j = 0, 1, \ldots, M-2)$) plus the c unknown constants a_k. However $M \times (N+1) - 1$ of these equations are linearly independent, since the generator matrix of the Markov process is singular.

On the other hand, an additional independent equation is provided by (2.8). Hence, the number of independent linear equations is $M \times (N+1)$.

Clearly, this set of $M \times (N+1)$ independent linear equations with $(M-1) \times (N+1)+c$ unknowns has a unique solution if, and only if, $M \times (N+1) = (M-1) \times (N+1) + c$. Or, that condition is $c = N+1$. Here, we make the following hypothesis:

- If $c > N+1$, there are more unknowns than the the number of equations to be satisfied. That means infinitely many different solutions of the balance equations are possible and hence many different steady states. Since, in reality, the process can have only a unique steady state if it is stable, it can be said that c can not be greater than $N+1$. On the other hand, if $c < N+1$, a feasible solution does not exist for the balance equations and hence that corresponds to the instability of the process.

In fact, this is what has been found in all our examples. This analysis, together with the fact that an irreducible Markov process has a steady-state distribution if, and only if its balance and normalisation equations have a unique solution, implies

Proposition 1 *The condition $c = N+1$ (the number of eigenvalues of $Q(\lambda)$ strictly inside the unit disk is equal to the number of operative states of the Markov process), is necessary and sufficient for the ergodicity of the Markov process X.*

It is interesting to note, from proposition 1, as long as the process X is irreducible, the stability or ergodicity of this process is determined purely by the matrices A, B and C, and does not depend on the j-dependent matrices A_j, B_j and C_j. This is because the eigenvalues and eigenvectors depend purely on A, B and C.

2.4 Spectral analysis

In this section the nature of the eigenvalues of $Q(\lambda)$ is examined. The eigenvalues and eigenvectors of $Q(\lambda)$ satisfy,

$$\psi[Q_0 + Q_1\lambda + Q_2\lambda^2] = 0, \tag{2.16}$$

We get the following results.

Theorem 1 *If Q_0 is non-singular, then $Q(\lambda)$ does not have zero-eigenvalues.*

Proof Let $\lambda = 0$, then we get, from (2.11), $det[Q(\lambda)] = det[Q_0] \neq 0$. Hence, the result. ◇

Theorem 2 *If Q_0 is singular with a rank $N+1-r_0$, then (i) $Q(\lambda)$ has zero-eigenvalues of multiplicity r_0 if the sum of all principal minors of Q_0 of order $N+1-r_0$ is non-zero. (ii) it has r_0 corresponding independent eigenvectors.*

Proof Let $\lambda = 0$ in (2.16), then we get $\psi Q_0 = 0$. Hence, the zero-eigenvalue of $Q(\lambda)$ and the zero-eigenvalue of Q_0 have the same multiplicity. Now, refer to Theorem 2.1.2, Lemma of [37, pages 54-57] to complete the proof of part (*i*), and Definition (a) of [37, page 21], Theorem 1.16.2 of [37, page 45] for part (*ii*). ◇

Theorem 3 *If Q_2 is non-singular, then $Q(\lambda)$ has $2N+2$ eigenvalues.*

Proof Multiply the equation (2.16) on right by Q_2^{-1}, then it becomes

$$\psi[T_0 + T_1\lambda + I\lambda^2] = 0,$$

where $T_0 = Q_0Q_2^{-1}$, $T_1 = Q_1Q_2^{-1}$ and I is the Unit matrix. By introducing the auxiliary vector $\varphi = \lambda\psi$, this equation can be rewritten in the equivalent linear form as,

$$[\psi \quad \varphi] \begin{bmatrix} 0 & -T_0 \\ I & -T_1 \end{bmatrix} = \lambda[\psi \quad \varphi] \,. \tag{2.17}$$

This linear eigenvalue problem has $2N+2$ finite solutions for λ [37]. Every (λ, ψ) that satisfies equation (2.17) does satisfy equation (2.16), and the converse also is true. Hence, $Q(\lambda)$ has $2N+2$ eigenvalues. Note that the square matrix involved in equation (2.17) is simple iff multiple eigenvalues would have independent multiple eigenvectors. ◇

Theorem 4 *If Q_0 is non-singular and Q_2 is singular with rank $N+1-r_2$, then the number of eigenvalues of $Q(\lambda)$ is $2N+2-r_2$.*

Proof Since Q_0 is non-singular, by Theorem 1, $\lambda = 0$ is not an eigenvalue of $Q(\lambda)$. Let us, then, introduce the auxiliary variable $\gamma = 1/\lambda$ and, transform the variable to γ, by substitution. Then, we get,

$$\psi[Q_2 + Q_1\gamma + Q_0\gamma^2] = 0. \tag{2.18}$$

For every (λ, ψ) satisfying (2.16), there exists a $\gamma = 1/\lambda$, such that (γ, ψ) satisfies the new quadratic eigenvalue problem (2.18). And, for every $(\gamma \neq 0, \psi)$ satisfying (2.18), there exists a $\lambda = 1/\gamma$, such that (λ, ψ) satisfies (2.16). Hence, the eigenvalues of (2.16) are the non-zero eigenvalues of (2.18). Applying Theorems 2, 3, in the case of (2.18), it can be said the non-zero eigenvalues of (2.18) are indeed, $2N+2-r_2$. Hence, the result. ◇

Theorem 5 *If Q_0 is singular with a rank $N+1-r_0$ and Q_2 is singular with a rank $N+1-r_2$, then $Q(\lambda)$ has (i) r_0 zero-eigenvalues, and (ii) $2N+2-r_0-r_2$ non-zero eigenvalues.*

Proof Since Q_0 is singular, applying Theorem 2, part (*i*) is proved. In order to prove (*ii*), introduce an auxiliary variable $\gamma = \frac{(\theta+\lambda)}{(\theta-\lambda)}$, where θ is arbitrary. θ can be so chosen that, for every λ from a finite set of values, the corresponding γ exists. λ can be expressed in terms of γ as,

$$\lambda = \frac{\theta(\gamma-1)}{(\gamma+1)} \tag{2.19}$$

Substituting this in (2.16), we get

$$\psi[(Q_0 - \theta Q_1 + \theta^2 Q_2) + 2(Q_0 - \theta Q_2)\gamma + (Q_0 + \theta Q_1 + \theta^2 Q_2)\gamma^2] = 0 \,. \qquad (2.20)$$

The parameter θ is so chosen that $(Q_0 + \theta Q_1 + \theta^2 Q_2)$ is non-singular($\theta \neq 0, 1$). Applying Theorem 3, we can say that equation (2.20) has $2N + 2$ eigenvalues. Also, it can be seen by substitution, of those $2N+2$ eigenvalues r_2 are multiple eigenvalues at $\gamma = -1$ and r_0 are multiple eigenvalues at $\gamma = 1$.

Now, for every (λ, ψ) satisfying (2.16), there exists a unique (γ, ψ) that satisfies (2.20). Also, for every $(\gamma \neq -1, \psi)$, there exists a corresponding unique (λ, ψ) satisfying (2.16). Hence, the λ's corresponding to the γ's that are not equal to -1, are all the eigenvalues of (2.16). When $\gamma = 1$, the corresponding $\lambda = 0$. Hence, the result. $\diamond$

2.5 Further notes on stability and spectral analysis

There is another way of looking at the stability problem, though it is a conjecture and not a rigorous proof. Let $q_0, q_1, \ldots, q_N$ be the marginal steady state probabilities of the operative states. Then,

$$q_i = \sum_{j=0}^{\infty} p_{i,j}$$

Also, let the row vector $\mathbf{v}$ be

$$\mathbf{v} = (q_0, q_1, \ldots, q_N) = \sum_{j=0}^{\infty} \mathbf{v}_j$$

Now, consider the process X is stable, but is driven to the *verge of instability*. This can be done by tuning A, B and C accordingly. In this state, it is possible to assume that the process, still stable, yet only negligibly rarely visits the states that are below $j = M$. This must be true since, otherwise we can not term this as at the verge of instability. In this state, a simple expression is possible for the probabilities, q_i, given by,

$$\mathbf{v}(A - D^A + B - D^B + C - D^C) = 0 \; ; \; \mathbf{ve} = 1 \; . \tag{2.21}$$

The matrix $(A - D^A + B - D^B + C - D^C)$ is singular and normally of rank N. Hence, equations (2.21) can be solved uniquely for $\mathbf{v}$.

Also, since the Markov process X is stable, we also have,

$$q_i \geq 0 \; ; \; i = 0, 1, \ldots, N \; . \tag{2.22}$$

Since, in this state, the process is assumed to dwell in the region, $j \geq M$, the average drift up by which the process moves upwards and the average drift down by which the process moves downwards are $\mathbf{v}B\mathbf{e}$ and $\mathbf{v}C\mathbf{e}$ respectively. Since, X is stable, the average drift up should be strictly less the average drift down. Hence, we get,

$$\mathbf{v}B\mathbf{e} < \mathbf{v}C\mathbf{e} \; . \tag{2.23}$$

Thus, we have arrived at an alternate stability condition defined by the equations (2.21, 2.22), together with the inequality condition (2.23).

We feel this is a necessary condition, and not sure if it is also sufficient. Work is underway to prove mathematically, if this condition is necessary, or sufficient, or both. This condition may be equivalent to the proposition 1. Also, it is worth to notice this condition also does not depend on the j-dependent A_j, B_j and C_j.

It is to be noted that $\lambda = 1$ satisfies the equation (2.11) and hence it is an eigenvalue for all possible values of A, B and C. Numerical experiments have shown that an ergodic process may exhibit complex conjugate eigenvalues inside the unit disk. However, the eigenvalue with the largest modulus less than 1 is always real(it is the Perron-Frobenius eigenvalue of Neuts' matrix R [51] of the matrix-geometric method). When the parameters are varied in such a way the inequality (2.23) is violated, at the point of violation that eigenvalue leaves the unit circle, passing through the point 1. Here are two eigenvalues equal to 1, exactly at the point of violation. ◇

2.5.1 The dual process $\overline{X}$

It is interesting, here, to examine a related irreducible process, $\overline{X}$, which we call the *dual* of the process X. The j-independent upward transition rates in X are the j-independent downward transition rates in $\overline{X}$, and the j-independent downward transition rates in X are the j-independent upward transition rates in $\overline{X}$. The j-independent lateral transitions are the same in both the processes. $\overline{X}$ is obtained by simply interchanging the matrices B and C in X. The j-dependent transition rate matrices of $\overline{X}$ can be quite arbitrary, but ensuring the irreducibility of the process. The characteristic matrix polynomial of $\overline{X}$ is then given by

$$\overline{Q}(\beta) = Q_2 + Q_1\beta + Q_0\beta^2 \tag{2.24}$$

The eigenvalues and eigenvectors of $\overline{Q}(\beta)$ satisfy

$$\phi\overline{Q}(\beta) = 0 \; ; \; det[\overline{Q}(\beta)] = 0 \; . \tag{2.25}$$

It can be shown as before that, $\overline{X}$ is ergodic if and only if $\overline{Q}(\beta)$ has $N+1$ number of eigenvalues inside its unit circle, and it can not have greater than $N+1$ eigenvalues inside its unit circle.

It can now be conjuctured that between X and $\overline{X}$, if one is stable, automatically the other is unstable. (Though it is possible that both X and $\overline{X}$ are unstable, this might happen, for example, when the inequality (2.23) becomes equality. And, there may be other situations in which both X and $\overline{X}$ are unstable.) That is so, because the equations (2.21) for $\mathbf{v}$ are the same for both the processes but the stability condition for $\overline{X}$ is exactly the opposite of (2.23). If (2.23) is satisfied X is stable and $\overline{X}$ is unstable. If (2.23) is violated and the opposite of it is valid then the reverse is true. ◇

There is a close relationship between the eigenvalue-eigenvectors of $Q(\lambda)$ and those of $\overline{Q}(\beta)$. For every non-zero eigenvalue-eigenvector pair, (λ, ψ), of $Q(\lambda)$, there exist a unique $\beta = 1/\lambda$ and a $\phi = \psi$ that satisfy (2.25), hence that (β, ϕ) is an eigenvalue-eigenvector pair of $\overline{Q}(\beta)$. Thus, the number of non-zero eigenvalues of $Q(\lambda)$ is the same as those of $\overline{Q}(\beta)$.

The following three statements are proved earlier, and also numerically verified by limited examples:

(a) If Q_2 is non-singular then $Q(\lambda)$ has $2N + 2$ number of eigenvalues. If Q_0 is non-singular then $\overline{Q}(\beta)$ has $2N + 2$ number of eigenvalues.

(b) If Q_0 has a rank $N + 1 - r_0$ then $Q(\lambda)$ has r_0 number of zero-eigenvalues [37], that would also correspond to $\overline{Q}(\beta)$ having $2(N + 1) - r_0$ number of eigenvalues. On the other hand, if Q_2 has a rank $N + 1 - r_2$ then $\overline{Q}(\beta)$ will have r_2 number of zero-eigenvalues which implies $Q(\lambda)$ will have $2(N+1) - r_2$ number of eigenvalues.

(c) Thus, if Q_0 has a rank $N + 1 - r_0$ and Q_2 has a rank $N + 1 - r_2$ then, $Q(\lambda)$ will have r_0 number of zero-eigenvalues and $2(N + 1) - r_0 - r_2$ number of non-zero eigenvalues, whereas $\overline{Q}(\beta)$ will have r_2 number of zero-eigenvalues and $2(N + 1) - r_0 - r_2$ number of non-zero eigenvalues.

In the following analysis, assume Q_0 has a rank $N + 1 - r_0$ and Q_2 has a rank $N+1-r_2$, where $0 \leq r_0, r_2 \leq N+1$. Then, if X is stable, the number of eigenvalues within its unit circle is $N + 1$, exactly on the unit circle is one($\lambda = 1.0$), and the remaining eigenvalues, $N - r_2$ in number, are outside the unit circle. The reciprocals of these last $N - r_2$ eigenvalues are the non-zero eigenvalues of $\overline{Q}(\beta)$ within its unit circle. $\overline{Q}(\beta)$ also has r_2 number of zero-eigenvalues. Thus, total number of eigenvalues of $\overline{Q}(\beta)$ that are strictly within its unit circle is $N - r_2 + r_2 = N$. Hence,

$\overline{X}$ is unstable. It is also possible to prove, in similar lines, that if $\overline{X}$ is stable that implies X is unstable. Now, we have,

Proposition 2 *Between the irreducible processes X and $\overline{X}$, if one is ergodic that automatically implies that the other is unstable.*

In order to arrive at another result, let X have c number of eigenvalues inside its unit circle. There is one eigenvalue exactly on the unit circle. Thus, the number of eigenvalues strictly outside the unit circle is $2N+2-r_2-1-c$. Hence, in the case of $\overline{X}$, the number of eigenvalues strictly within its unit circle is, r_2 zero-eigenvalues plus $2N+2-r_2-1-c$ non-zero ones. That is a total of $2N+2-1-c$ eigenvalues strictly inside the unit circle of $\overline{Q}(\beta)$. We have already hypothesized that this number can not be greater than $N+1$. That is possible only if c is not less than N. Hence, the following assertion,

Proposition 3 *Given that X is irreducible and a unique non-negative solution exists for (2.21), X is stable if and only if its characteristic polynomial $Q(\lambda)$ has exactly $N+1$ eigenvalues strictly within the unit circle, and, X is unstable if and only if $Q(\lambda)$ has exactly N eigenvalues strictly within the unit circle.*

2.6 Computation

However, the computational aspects are not quite trivial. The spectral expansion procedure consists of the following steps:

(1) Compute the eigenvalues, λ_k, and the corresponding eigenvectors, ψ_k, of $Q(\lambda)$. If $c = N$, stop; a steady-state distribution does not exist. If $c = N+1$, the process X is ergodic and a steady-state distribution does exist; proceed to step 2. If $c \neq N+1$ and $c \neq N$, then stop; the eigenvalue computation is not accurate.

(2) Obtain $\mathbf{v}_{M-1}$ and $\mathbf{v}_M$ from the equation (2.14) in terms of the coeffients a_k. Now, solve the finite set of linear equations (2.6) and (2.8) for the constants a_k and the vectors $\mathbf{v}_j$, $j \leq M-2$. The solution is done.

(3) Use the obtained solution for the purpose of determining various moments, marginal distributions, percentiles and other system performance measures that are of interest.

Step 1 is performed as follows: The 'brute force' approach which relies on first evaluating the scalar polynomial $det[Q(\lambda)]$, then finding its roots may be computationally inefficient and very much time consuming. An alternative which is preferable in most cases is to reduce the quadratic eigenvalue-eigenvector problem

$$\psi[Q_0 + Q_1\lambda + Q_2\lambda^2] = 0, \tag{2.26}$$

to a linear one of the form $\mathbf{y}Q = \lambda\mathbf{y}$, where Q is a matrix of size $(2N+2)\times(2N+2)$. Efficient and highly accurate techniques such as the QR algorithm or the QZ algorithm, exist for the solution of the the latter problem.

This linearisation can be achieved quite easily when at least one of the matrices B and C is non-singular. Indeed, suppose that $C^{-1} = Q_2^{-1}$ exists. After multiplying (2.26) on the right by Q_2^{-1}, it becomes

$$\psi[T_0 + T_1\lambda + I\lambda^2] = 0 \ , \tag{2.27}$$

where $T_0 = Q_0Q_2^{-1}$ and $T_1 = Q_1Q_2^{-1}$. I is the unit matrix. By introducing the vector $\varphi = \lambda\psi$, equation (2.26) can be rewritten in the equivalent linear form as

$$[\psi \quad \varphi]\begin{bmatrix} 0 & -T_0 \\ I & -T_1 \end{bmatrix} = \lambda[\psi \quad \varphi] \ . \tag{2.28}$$

The number of eigenvalues of that linear form is $2N+2$ and hence those of (2.26) should be $2N+2$ if Q_2 is non-singular.

If $C = Q_2$ is singular but $B = Q_0$ is not, then it is proposed to solve the equation (2.25) for β's and the corresponding ϕ's, following the above procedure. Since Q_0 is non-singular the number of β's would be $2N+2$ of which r_2 are zeroes and others non-zero. The non-zero β's, $2N+2-r_2$ in number, can then be inverted to get the non-zero λ's of (2.26). Since Q_0 is non-singular $Q(\lambda)$ can not have zero-eigenvalues. Hence, the total eigenvalues of $Q(\lambda)$, in this case, has to be $2N+2-r_2$.

If both Q_0 and Q_2 are singular, then the desired result is achieved by introducing an auxiliary variable [31] $\gamma = (\theta+\lambda)/(\theta-\lambda)$ by substituting the following in (2.26),

$$\lambda = \frac{\theta(\gamma-1)}{(\gamma+1)} \tag{2.29}$$

Then, we get

$$\psi[(Q_0 - \theta Q_1 + \theta^2 Q_2) + 2(Q_0 - \theta Q_2)\gamma + (Q_0 + \theta Q_1 + \theta^2 Q_2)\gamma^2] = 0 \ . \tag{2.30}$$

Also let $S = Q_0 + \theta Q_1 + \theta^2 Q_2$, the parameter θ is arbitrary and it is so chosen that S is non-singular($\theta \neq 0, 1$). It is now easy to see that for every λ satisfying (2.26), there exists a γ satisfying (2.30). Hence, it is possible to get all the eigenvalues of $Q(\lambda)$ when all the eigenvalues of (2.30) are known. Let $T_2 = (Q_0 - \theta Q_1 + \theta^2 Q_2)S^{-1}$, $T_3 = 2(Q_0 - \theta^2 Q_2)S^{-1}$ and $\varphi = \gamma\psi$. Then, we get

$$[\psi \quad \varphi] \begin{bmatrix} 0 & -T_2 \\ I & -T_3 \end{bmatrix} = \gamma[\psi \quad \varphi] \tag{2.31}$$

The equation (2.31) is solved for all the γ's and ψ's. λ's are then derived from γ's, by equation (2.29). It can be seen that equation (2.30) has a total of $2N+2$ eigenvalues, of them r_0 are multiple eigenvalues at $\gamma = 1$, r_2 are multiple eigenvalues at $\gamma = -1$. When $\gamma = 1$ that corresponds to $\lambda = 0$, from equation (2.29). And, when $\gamma = -1$, a corresponding λ does not exist, and for every other value of γ, a corresponding λ does exist. Hence, $Q(\lambda)$ will have a total of $2N+2-r_2$ eigenvalues of which r_0 are zero-eigenvalues.

Step 2 is to compute $M(N+1)$ unknowns, some of them complex, from $M(N+1)$ independent linear simultaneous equations. It can be expensive if M is large. Some simplification is possible. From (2.14), $\mathbf{v}_M$ and $\mathbf{v}_{M-1}$ can be expressed in terms of the $N+1$ constants, a_k. If the matrices B_j $(j = 0, 1, \ldots, M-2)$ are non-singular, then it is possible to express the vectors $\mathbf{v}_{M-2}, \mathbf{v}_{M-3}, \ldots, \mathbf{v}_0$ in terms of $\mathbf{v}_{M-1}$ and $\mathbf{v}_M$, and thus in terms of a_k's, using the equations (2.6) for $(j = M-1, M, \ldots, 1)$. One is then left with the vector equation (2.6) for $j = 0$, plus (2.8). These are just a total of $N+1$ independent linear simultaneous equations for $N+1$ unknowns, a_k's.

If l is the number of real eigenvalues and m is the number of complex eigenvalues, strictly inside the unit circle, of $Q(\lambda)$ then, $\mathbf{v}_j$ can also be expressed as,

$$\mathbf{v}_j = \sum_{k=0}^{l-1} a_{r,k}\boldsymbol{\psi}_{r,k}\lambda_{r,k}^{j-M+1} + \sum_{k=0}^{m-1} a_{c,k}\boldsymbol{\psi}_{c,k}\lambda_{c,k}^{j-M+1} \ , \tag{2.32}$$

where $a_{r,k}$, $\boldsymbol{\psi}_{r,k}$, $\lambda_{r,k}$ $(k = 0, 1, \ldots, l-1)$ are real and $a_{c,k}$, $\boldsymbol{\psi}_{c,k}$, $\lambda_{c,k}$ $(k = 0, 1, \ldots, m-1)$ are complex. Number of unknowns are, l real and m complex$(l+m = N+1)$, amounting to $l+2m$ real unknowns. m is either 0 or even and the complex eigenvalue-eigenvectors occur in conjugates. That is to say, by suitably numbering, $\lambda_{c,1} = \lambda_{c,0}^*$, $\boldsymbol{\psi}_{c,1} = \boldsymbol{\psi}_{c,0}^*$; $\lambda_{c,3} = \lambda_{c,2}^*$, $\boldsymbol{\psi}_{c,3} = \boldsymbol{\psi}_{c,2}^*$; $\ldots$; $\lambda_{c,m-1} = \lambda_{c,m-2}^*$, $\boldsymbol{\psi}_{c,m-1} = \boldsymbol{\psi}_{c,m-2}^*$. Also, $a_{c,1} = a_{c,0}^*$, $a_{c,3} = a_{c,2}^*$, $\ldots$, $a_{c,m-1} = a_{c,m-2}^*$. Equation (2.32) can then be modified as,

$$\mathbf{v}_j = \sum_{k=0}^{l-1} x_k\boldsymbol{\psi}_{r,k}\lambda_{r,k}^{j-M+1} + 2\sum_{k=0}^{m/2-1} [x_{l+2k}Re(\boldsymbol{\psi}_{c,2k}\lambda_{c,2k}^{j-M+1}) - x_{l+2k+1}Im(\boldsymbol{\psi}_{c,2k}\lambda_{c,2k}^{j-M+1})] \tag{2.33}$$

in which, x_k are $N+1$ unknowns that are real, Re stands for the real part of its succeeding term in brackets and Im stands for the imaginary part. The coefficients x_k are related to $a_{r,k}$ and $a_{c,k}$ as: $x_k = a_{r,k}$ for $k = 0, 1, \ldots, l-1$. And, $x_l = Re(a_{c,0})$, $x_{l+1} = Im(a_{c,0})$; $x_{l+2} = Re(a_{c,2})$, $x_{l+3} = Im(a_{c,2})$; $\ldots$; $x_{l+m-2} = Re(a_{c,m-1})$, $x_{l+m-1} = Im(a_{c,m-1})$. The problem is now reduced to finding $N+1$ real unknowns by solving $N+1$ independent linear simultaneous equations.

When x_k's are known, the required performance measures of the system can be computed quite easily with aid of (2.33).

2.7 Generalisations

Several generalisations and extensions of the model described above are possible. An extension is presented in a later chapter. That is on extending spectral expansion solution to Markov processes of finite state space. A generalisation is taken up in this section. This generalisation incorporates bounded multi-step jumps in J, in either direction.

2.7.1 Multi-step jumps in J

This leads to a Markov process $Y = \{[I(t), J(t)]; t \geq 0\}$, on the state space $\{0, 1, \ldots, N\} \times \{0, 1, \ldots\}$, where the variable $J(t)$ may jump by arbitrary, but bounded amounts in either direction. Y evolves due to the following possible transitions:

(a) A_j - purely lateral transitions - From state (i, j) to state (k, j) $(0 \leq i, k \leq N\ ;\ i \neq k\ ;\ j = 0, 1, \ldots)$;

(b) $B_{j,s}$ - bounded s-step upward transitions - From state (i, j) to state $(k, j+s)$ $(0 \leq i, k \leq N\ ;\ 1 \leq s \leq y_1\ ;\ y_1 \geq 1\ ;\ j = 0, 1, \ldots)$;

(c) $C_{j,s}$ - bounded s-step downward transitions - From state (i, j) to state $(k, j-s)$ $(0 \leq i, k \leq N\ ;\ s \leq j\ ;\ 1 \leq s \leq y_2\ ;\ y_2 \geq 1\ ;\ j = 1, 2, \ldots)$.

A_j, $B_{j,s}$ $(s = 1, 2, \ldots, y_1)$ and $C_{j,s}$ $(s = 1, 2, \ldots, y_2)$ are the transition rate matrices associated with (a),(b) and (c) respectively. $C_{j,s} = 0$ if $j < s$. We assume there is a threshold M, $M \geq y_1$, such that

$$A_j = A, j \geq M\ ;\ B_{j,s} = B_s, j \geq M - y_1\ ;\ C_{j,s} = C_s, j \geq M\ ; \tag{2.34}$$

for the possible values of s.

Defining again the diagonal matrices D_j^A, $D_{j,s}^B$ and $D_{j,s}^C$, whose i^{th} diagonal element is equal to the i^{th} row sum of A_j, $B_{j,s}$ and $C_{j,s}$, respectively. Then the balance equations are,

$$\mathbf{v}_j[D_j^A+\sum_{s=1}^{y_1} D_{j,s}^B+\sum_{s=1}^{y_2} D_{j,s}^C] = \sum_{s=1}^{y_1} \mathbf{v}_{j-s}B_{j-s,s}+\mathbf{v}_jA_j+\sum_{s=1}^{y_2} \mathbf{v}_{j+s}C_{j+s,s} \ ; \ j = 0, 1, \ldots, M-1 \tag{2.35}$$

It is assumed $\mathbf{v}_{j-s} = 0$ if $j < s$. The corresponding j-independent set is,

$$\mathbf{v}_j[D^A + \sum_{s=1}^{y_1} D_s^B + \sum_{s=1}^{y_2} D_s^C] = \sum_{s=1}^{y_1} \mathbf{v}_{j-s}B_s + \mathbf{v}_jA + \sum_{s=1}^{y_2} \mathbf{v}_{j+s}C_s \ , \ j \geq M \tag{2.36}$$

where D^A, D_s^B and D_s^C are the j-independent versions of the diagonal matrices, D_j^A, $D_{j,s}^B$ and $D_{j,s}^C$ respectively.

In addition, we have for the sum of all probabilities,

$$\sum_{j=0}^{\infty} \mathbf{v}_j\mathbf{e} = 1.0 \, . \tag{2.37}$$

As before, (2.36) can be rewritten as a vector difference equation, of order $y = y_1 + y_2$, with constant coefficients:

$$\sum_{k=0}^{y} \mathbf{v}_{j+k}Q_k = 0 \ ; \ j \geq M - y_1 \tag{2.38}$$

Here, $Q_k = B_{y_1-k}$ for $k = 0, 1, \ldots, y_1 - 1$; $Q_{y_1} = A - D^A - \sum_{s=1}^{y_1} D_s^B - \sum_{s=1}^{y_2} D_s^C$; and $Q_k = C_{k-y_1}$ for $k = y_1 + 1, y_1 + 2, \ldots, y_1 + y_2$.

The corresponding matrix polynomial is,

$$Q(\lambda) = \sum_{k=0}^{y} Q_k\lambda^k \, . \tag{2.39}$$

The development, hence forth, is similar to that in section 2.3. The normalizeable solution of equation (2.36) is of the form,

$$\mathbf{v}_j = \sum_{k=0}^{c-1} a_k\psi_k\lambda_k^{j-M+y_1} \ ; \ j = M - y_1, M - y_1 + 1, \ldots \tag{2.40}$$

where λ_k are all the eigenvalues, c in number, of $Q(\lambda)$ strictly inside the unit disk ψ_k are the corresponding independent eigenvectors, and a_k are the arbitrary constants $(k = 0, 1, \ldots, c-1)$. The latter constants are determined with the aid of the state-dependent balance equations, that is (2.35) for $j \le M-1$, and the normalisation equation. In this case, following the same line of thought as in section 2.6., it is possible to find the value of c that is necessary and sufficient condition for the ergodicity of Y. This time, the number of linear simultaneous equations is $M(N+1)$ from (2.35), and 1 from the (2.37) out of which $M(N+1)$ are linearly independent. And, we have the unknowns $(M-y_1)(N+1)$, the probabilities of states below $(j < M-y_1)$, and the c unknowns. These equations can have unique solution if and only if $(M-y_1)(N+1)+c = M(N+1)$, or, equivalently $c = y_1(N+1)$. Thus,

Proposition 4 *The condition the number of eigenvalues of $Q(\lambda)$ strictly inside the unit disk, $c = y_1(N+1)$, is necessary and sufficient for the ergodicity of the irreducible Markov process Y.*

It is worth having another look at the stability properties of Y. Let us define an irreducible process $\overline{Y}$, the *dual* process of Y. In $\overline{Y}$, the B_s $(s = 1, 2, \ldots, y_1)$ are the *downward* transition rate matrices and the C_s $(s = 1, 2, \ldots, y_2)$ are the *upward* transition rate matrices. Since the stability condition depends only on the j-independent matrices A, B_s and C_s, and it does not depend on the j-dependent versions, we need not go at length to define the j-dependent matrices for $\overline{Y}$.

Define the row vector, $\mathbf{v} = (q_0, q_1, \ldots, q_N)$. Following the same line of argument as in section 2.3, the necessary condition for the stability of Y is the existence of solution for the following equations,

$$\mathbf{v}[A - D^A + \sum_{s=1}^{y_1}(B_s - D_s^B) + \sum_{s=1}^{y_2}(C_s - D_s^C)] = 0 \ ; \ \mathbf{ve} = 0 \ ; \tag{2.41}$$

with the condition,

$$q_i \ge 0 \ ; \ i = 0, 1 \ldots, N \ ; \tag{2.42}$$

combined with the inequality condition,

$$\mathbf{v}[\sum_{s=1}^{y_1} sB_s]\mathbf{e} < \mathbf{v}[\sum_{s=1}^{y_2} sC_s]\mathbf{e} . \tag{2.43}$$

Similarly, the necessary condition for $\overline{Y}$ to be stable may be the existence of the solution for (2.41, 2.42), and the opposite of (2.43) being valid. Hence, we have

Proposition 5 *Between the irreducible processes Y and $\overline{Y}$, if one is ergodic that automatically implies that the other is unstable.*

Also, it is possible to write the characteristic polynomial $\overline{Q}(\lambda)$ of $\overline{Y}$, as

$$\sum_{k=0}^{y} Q_{y-k}\lambda^k ,$$

and analyse the relationship between the eigenvalue-eigenvectors of $Q(\lambda)$ and $\overline{Q}(\lambda)$. We may workout in similar lines as in section 2.3 and conclude,

Proposition 6 *Given Y is irreducible and a unique non-negative solution exists for (2.41), Y is stable if and only if its characteristic polynomial $Q(\lambda)$ has $y_1(N+1)$ number of eigenvalues strictly within the unit circle, and, Y is unstable if and only if $Q(\lambda)$ has exactly $y_1(N+1)-1$ eigenvalues inside the unit circle.*

For computational purposes, the polynomial eigenvalue-eigenvector problem of degree t can be transformed into a linear one in much the same way as in section 2.6. For example, suppose that Q_y is non-singular and multiply (2.38) on the right by Q_y^{-1}. That leads to the problem

$$\psi[\sum_{k=0}^{y-1} T_k\lambda^k + I\lambda^y] = 0 , \tag{2.44}$$

where $T_k = Q_kQ_y^{-1}$. Introducing the vectors $\varphi_k = \lambda^k\psi$, $k = 1, 2, \ldots, y-1$, we obtain the equivalent linear form

$$[\psi\ \varphi_1\ \cdots\ \varphi_{y-1}] \begin{bmatrix} 0 & & & -T_0 \\ I & 0 & & -T_1 \\ & \ddots & \ddots & \\ & & I & -T_{y-1} \end{bmatrix} = \lambda[\psi\ \varphi_1\ \cdots\ \varphi_{y-1}] . \tag{2.45}$$

Other transformations to linear form, quite similar to those in section 2.6, are possible when Q_y is singular and Q_0 is not, or when both Q_0 and Q_y are singular.

2.8 Conclusions

The mathematics of the spectral expansion method, in a general form, has been presented in this chapter. Further generalisation is also possible. For example, the spectral expansion solution is essentially for the equations (2.7). Hence, as long as these equations (2.7) are satisfied by the general solution, the probabilities below $j < M - 1$, can be more arbitrarily related than in equations (2.6) of section 2.3. Similar extension is possible for the model in section 2.7.

Several useful conclusions regarding the nature of the characteristic equation and its eigenvalue-eigenvectors are drawn. An efficient computation algorithm has been described in a good detail, this can serve as a guide to modellers interested in using the method. Accurate eigenvalue-eigenvectors are necessary since the performance measures can be very sensitive to them. The eigenvalue-eigenvector routines in the available packages such as NAG, seem to be quite good towards this end.

A generalisation covering multi-step jumps in the random variable $J(t)$, has been described and solved. Perhaps, it is possible to further explore the relationship between the processes X and $\overline{X}$, and, between Y and $\overline{Y}$. In particular, it is interesting to see under what conditions, these are unstable.

Several applications, extensions and analyses are presented in the later chapters.

2.9 Contributions

The contributions in this chapter are,

- Making use of the spectral expansion solution for $\mathbf{v}_j$, for the process X, an efficient procedure is developed for the computation of the steady state probabilities. Required steady state performance measures can be computed using these probabilities.
- The process Y is defined and its spectral expansion solution is formulated. Efficient computational procedure for steady state probabilities is are also developed.
- The spectral and stability analyses of the processes X and Y are developed.

Some of these contributions have appeared in our papers [4, 5].

Chapter 3

Examples

3.1 Introduction

The mathematics of the spectral expansion method has been described in a good detail in the previous chapter. This chapter is intended to illustrate how to apply this method to model and analyse given systems. This is done by taking two non-trivial examples. These systems are chosen quite different from each other, to demonstrate the use of spectral expansion for diverse of applications. The first example is a homogeneous multiprocessor system with unbounded queueing capacity, serving a stream of incoming jobs. The processors are prone to breakdowns from time to time, and failed processors are repaired. In such a sytem, the steady-state performance measures are computed using the formulae obtained in the previous chapter. The other example, is a series of two stages of service in tandem, with a finite intermediate waiting room, processing incoming requests sequentially in the same order.

3.2 A homogeneous multiprocessor system

The system under consideration is a homogeneous multiprocessor with N identical processors serving a common, unbounded job queue. This is illustrated in Figure 1. The processors break down from time to time. Single and independent failures of servers, as well as multiple and simultaneous failures are possible. Failed processors return to an operative state after being repaired. Single, independent repairs of processors as well as multiple, simultaneous repairs are allowed. When operative, each processor serves jobs, one at a time, and each job is allowed to occupy atmost one operative processor at a time. The policy concerning services interrupted by breakdowns is either *resume*, or *repeat with re-sampling*. Failure, repair and service times are assumed to be exponentially distributed. The system is illustrated in figure 1.

This sytem is modelled by spectral expansion. $I(t)$ is the *operative state* of the system, representing the number of operative processors at time t. $I(t)$ varies from 0 to N. $J(t)$ is the number of jobs in the system at time t, including those currently being served. The irreducible Markov process representing the system is given by, $X = \{[I(t), J(t)]; t \geq 0\}$, with state space $\{0, 1, \ldots, N\} \times \{0, 1, \ldots\}$. Jobs are assumed to arrive according to an independent Poisson process with rate σ_i when the operative state $I(t) = i$. Two kinds of failures are possible. One is, individual processors break down independently at rate ξ and are repaired independently at rate η. In addition, there can be 'global' simultaneous breakdowns of all currently operative processors, at rate ξ_0, and 'global' simultaneous repairs of all currently inoperative processors, at rate η_N. When a new job arrives or when a completed job departs from the system, the operative state does not change, unless there is an independent coincidence towards such a change. Hence, change in the operative state of the system is reflected only in the matrices A and A_j.

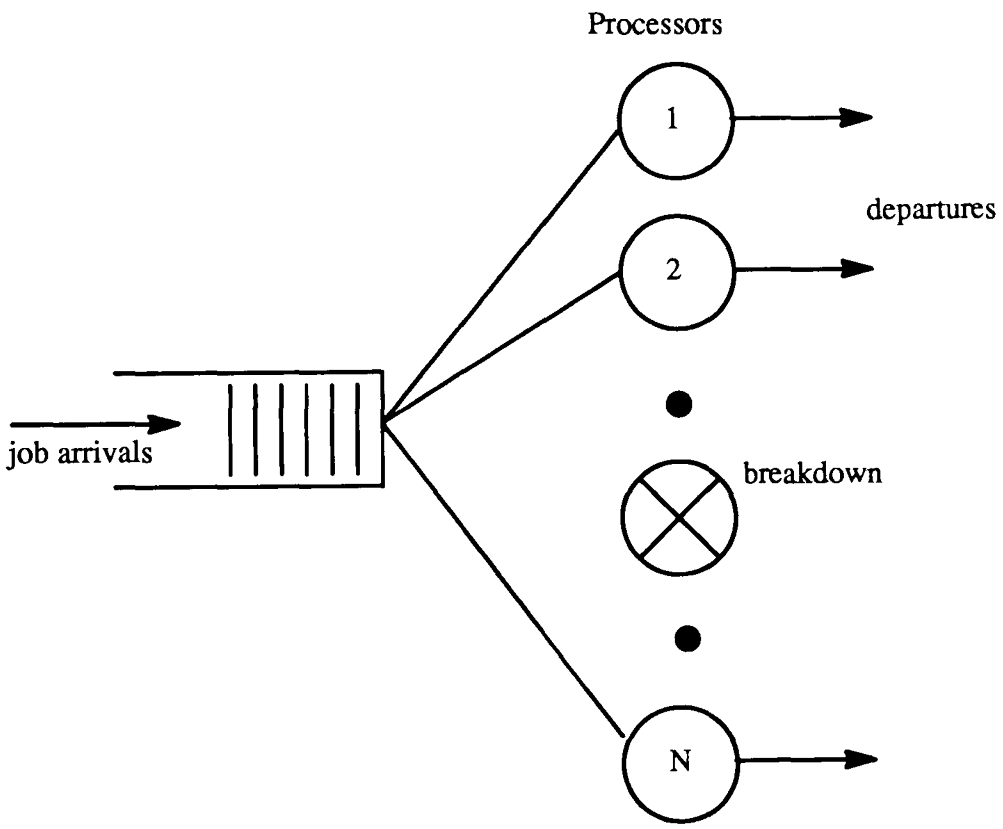

Figure 3.1: A homogeneous multiprocessor subject to breakdowns.

The one-step upward transitions are created by the arrivals of single jobs. Hence. B and B_j, the one-step upward transition rate matrices are given by,

$$B = B_j = diag[\sigma_0, \sigma_1, \ldots, \sigma_N] \tag{3.1}$$

This is valid for all values of j ($j = 0, 1, \ldots$).

The one-step downward transitions take place by the departures of single jobs, after their service completion. C and C_j are the one-step downward transition rate matrices. Let μ be the service rate of an operative processor. The departure rate of jobs at a time t depends on $I(t) = i$ and $J(t) = j$, and that is given by $C_j(i,i)$. If $i > j$, then every job has a processor for getting service, and not all operative processors are occupied, hence the departure rate, $C_j(i,i) = j\mu$. On the other hand, if $i \leq j$, then all the operative processors are occupied by jobs, hence the departure rate, $C_j(i,i) = i\mu$. Notice, $C_j(i,i)$ does not depend on j if $j \geq i$. Hence, C_j does not depend on j if $j \geq N$. So, we have,

$$\begin{aligned} C_j &= diag[0, min(j,1)\mu, min(j,2)\mu, \ldots, min(j,N)\mu] \; ; \; 0 < j < N \; , \\ C &= diag[0, \mu, 2\mu, \ldots, N\mu] \; ; \; j \geq N \; . \end{aligned} \tag{3.2}$$

C_0 is equal to 0.

In order to facilitate a comparison study, in addition to the performance evaluation, we take up three different types. Each of these three is distinguished by its breakdown and repair environment, i.e. by the form of the matrices A and A_j. These three environment types are:

- Type 1. In this environment, processors breakdown independently with a rate ξ per processor. Each inoperative processor is repaired at rate η. There are no simultaneous or correlated breakdowns or repairs, of multiple processors.

The matrices A_j and A are then given by,

$$A = A_j \; (j = 0, 1, \ldots) = \begin{bmatrix} 0 & N\eta & & & \\ \xi & 0 & (N-1)\eta & & \\ & 2\xi & 0 & \ddots & \\ & & \ddots & \ddots & \eta \\ & & & N\xi & 0 \end{bmatrix} . \tag{3.3}$$

- Type 2. Independent breakdowns and repairs as in Type 1, plus, 'global' breakdowns occuring at rate ξ_0. i.e, all currently operative processors break down together at this rate, apart from their independent breakdowns. Here:

$$A = A_j \; (j = 0, 1, \ldots) = \begin{bmatrix} 0 & N\eta & & & \\ \xi_0 + \xi & 0 & (N-1)\eta & & \\ \xi_0 & 2\xi & 0 & \ddots & \\ \vdots & & \ddots & \ddots & \eta \\ \xi_0 & & & N\xi & 0 \end{bmatrix} . \tag{3.4}$$

- Type 3. As in Type 2, but in addition, it is possible for all currently inoperative processors to be repaired simultaneously. This 'global' repair occurs at a rate η_N, on the inoperative processors:

$$A = A_j \; (j = 0, 1, \ldots) = \begin{bmatrix} 0 & N\eta & & & \eta_N \\ \xi_0 + \xi & 0 & (N-1)\eta & & \eta_N \\ \xi_0 & 2\xi & 0 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \eta + \eta_N \\ \xi_0 & & & N\xi & 0 \end{bmatrix} . \tag{3.5}$$

The threshold, M, is then given by $M = N$. And, the spectral expansion solution is valid from $j = M - 1 = N - 1$.

In these examples, since the operative state changes are caused only by the matrix A, it is possible to find simple expressions for the steady state distribution of the number of operative processors, and the total average service rate of the multiprocessor. Let v be the marginal distribution of the number of operative processors. Then,

$$\mathbf{v} = (q_0, q_1, \ldots, q_N) = \sum_{j=0}^{\infty} \mathbf{v}_j \,.$$

This is the stationary probability vector of the matrix $A - D^A$, and can be obtained by solving the equations,

$$\mathbf{v}(A - D^A) = 0 \; ; \; \mathbf{v}\mathbf{e} = 1 \,. \tag{3.6}$$

Then, the total average service rate, termed as the *service capacity* of the multiprocessor, is $\mathbf{v}C\mathbf{e}$ and the average incoming load is $\mathbf{v}B\mathbf{e}$. It can be asserted that the system is stable if, and only if, the average incoming load is less than the average service capacity:

$$\mathbf{v}B\mathbf{e} < \mathbf{v}C\mathbf{e} \,. \tag{3.7}$$

3.2.1 Performance measures and comparisons

Several sets of examples are considered to illustate the method. The results of each set is plotted in a figure. In order to make the comparisons as fair as possible, the parameters of the different systems that are compared, are chosen in such a way that, (*i*) they have the same number of processors of the same service rate, (*ii*) the service capacity of these, i.e. the right-hand side of (3.7), are the same. We have chosen to achieve this by judicious changes in the repair parameters. Thus, in Type 2, the introduction of global breakdowns ξ_0 is compensated by an appropriate increase in η; in Type 3, ξ_0 is compensated by a suitably chosen η_N, instead. Consequently, given identical arrival streams, the processing capacity and hence the ergodicity conditions, are the same for the three systems. Hence, these systems are comparable.

The arrival rates are independent of the operative state, in Figures 3.2-3.6. Systems with 10 processors are considered in Figures 3.2, 3.4, 3.5, 3.6, 3.7.

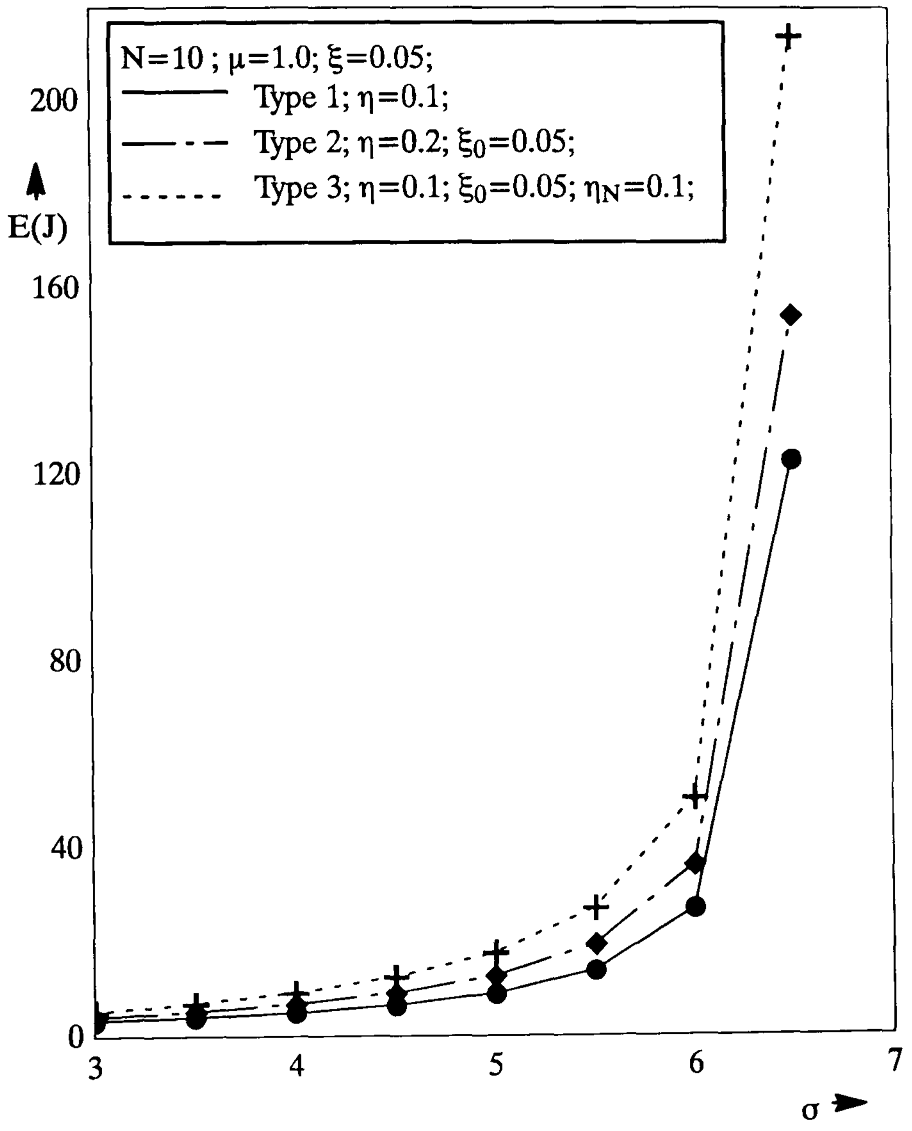

Figure 3.2: Average queue size as a function of the arrival rate. Service capacity=6.66667.

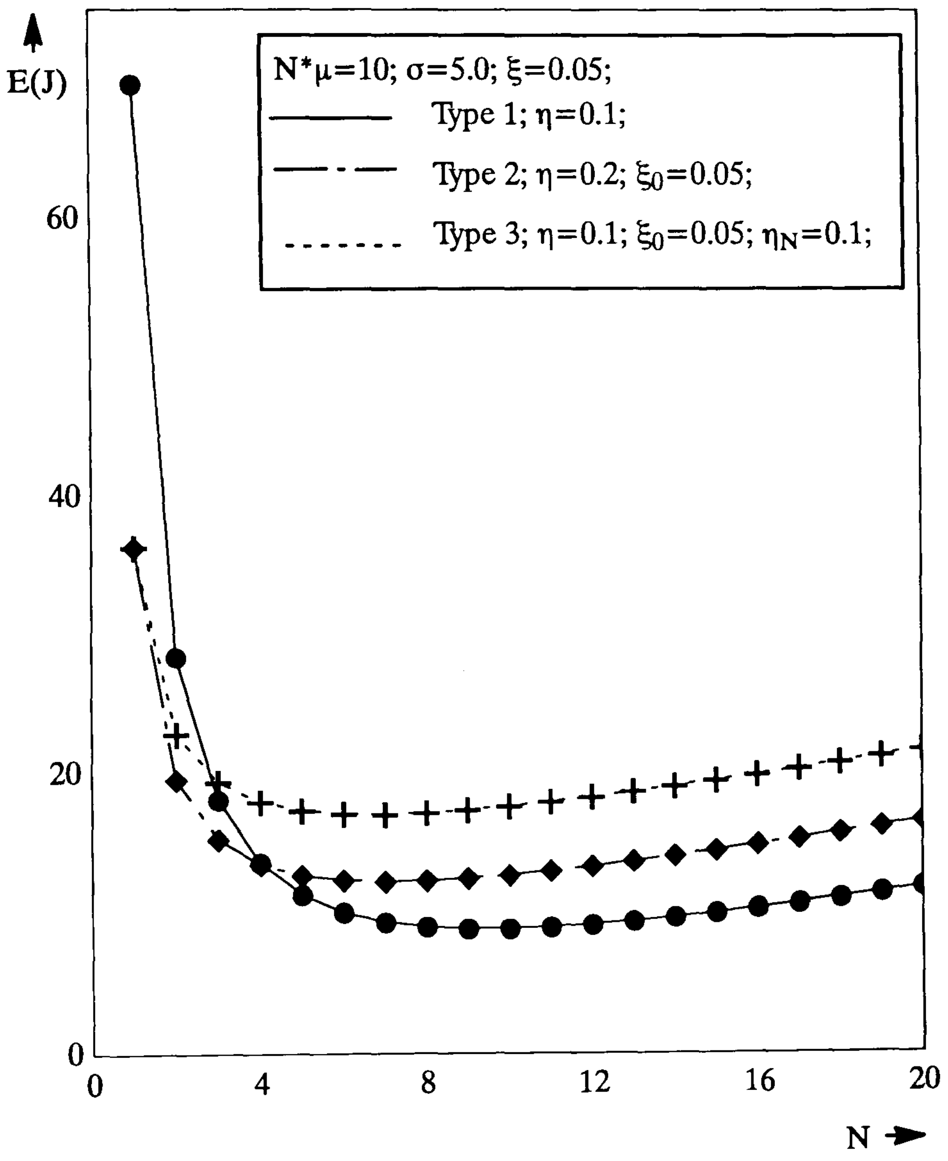

Figure 3.3: Average queue size as a function N, with $N\mu$ fixed. Service capacity=6.66667.

Figure 3.2 illustrates the effect of increasing the arrival rate, σ, on the average number of jobs, $E(J)$, in all the systems. In Types 2 and 3, the average failure rate and repair rate of a processor are higher than in Type 1. This fact reduces the variance of the total processing rate in Types 2 and 3, when compared to Type 1. At the same time, the correlated nature of failures in Types 2 and 3, and the correlated nature of repairs in Type 3 tend to increase the variance of the total processing rate of Syatem 3 in comparison to that in Type 1. In the examples in Figure 3.2, the latter factor dominates the former one. This results in an increased variance of the total processing rate in Type 3 over that of Type 2, and the latter over Type 1. Hence, the queue lengths in Type 1 are the lowest, and the queuelengths in Type 3 are highest.

In Figure 3.3, the number of processors and the service rate are varied together, so as to keep the product $N\mu$ fixed. The resulting graphs show that, for each system, there is an optimal number of processors that minimises the average queue size. That fact has been observed before [44], in the case of Type 1. The present results have two points of interest: First, for low values of N, Types 2 and 3 perform better than Type 1, whereas the reverse is true for large N. The likely explanation for this is that (i) increasing both the breakdown and repair rates tends to reduce the variance of processor availability, while (ii) increasing N tends to make it larger. Apparently, (i) is dominant for small N and (ii) is dominant for large N.

The second point of interest in Figure 3.3, is that the optimal value of N is larger for Type 1 than for Types 2 and 3. This seems counterintuitive, but perhaps the explanation lies again with the above trade-off.

Figures 3.4, 3.5 and 3.6 show conditional and unconditional 95-percentiles, together with $E(J)$, as functions of the arrival rate, for Types 1, 2 and 3, respectively.

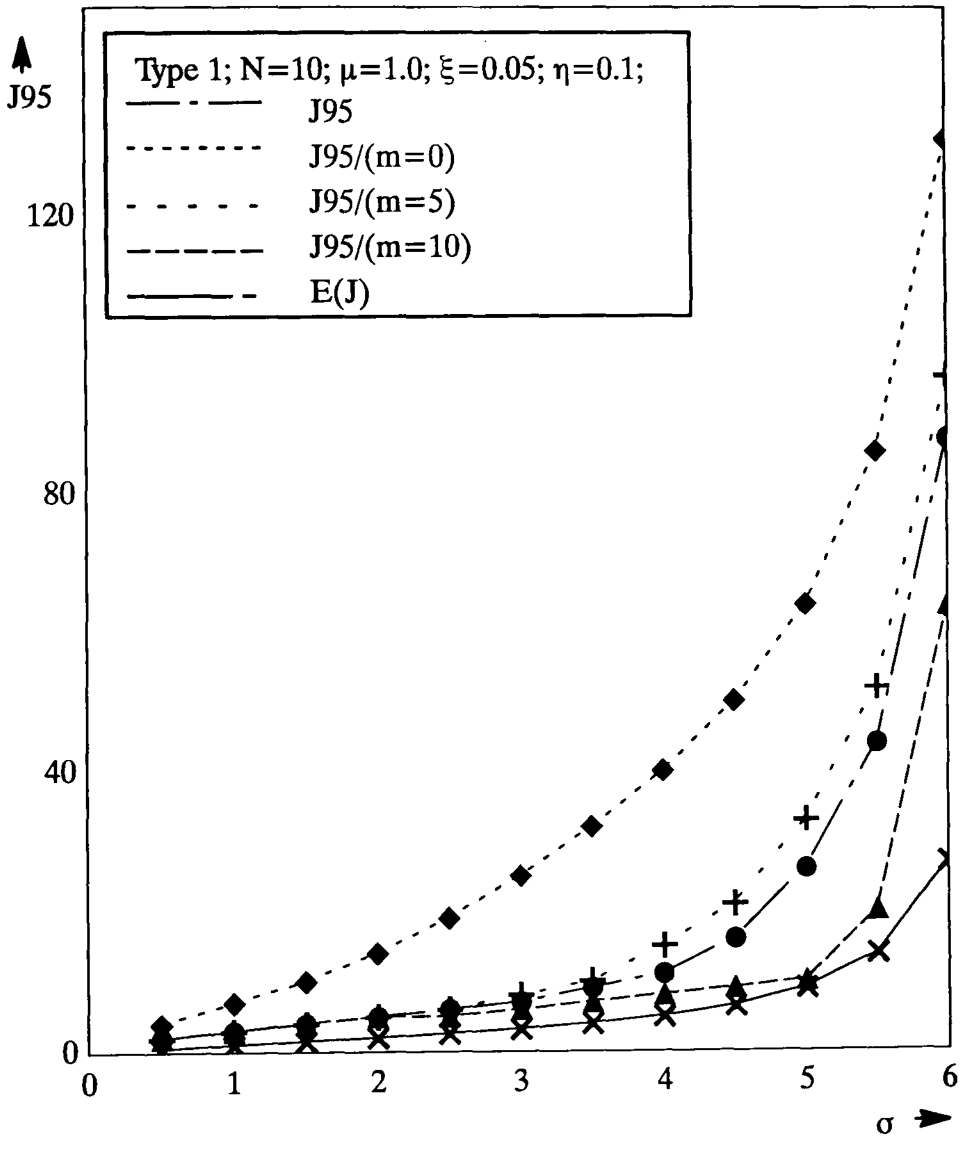

Figure 3.4: Conditional and unconditional 95-percentiles, and $E(J)$, as functions of the arrival rate, for Type 1. Service capacity=6.66667.

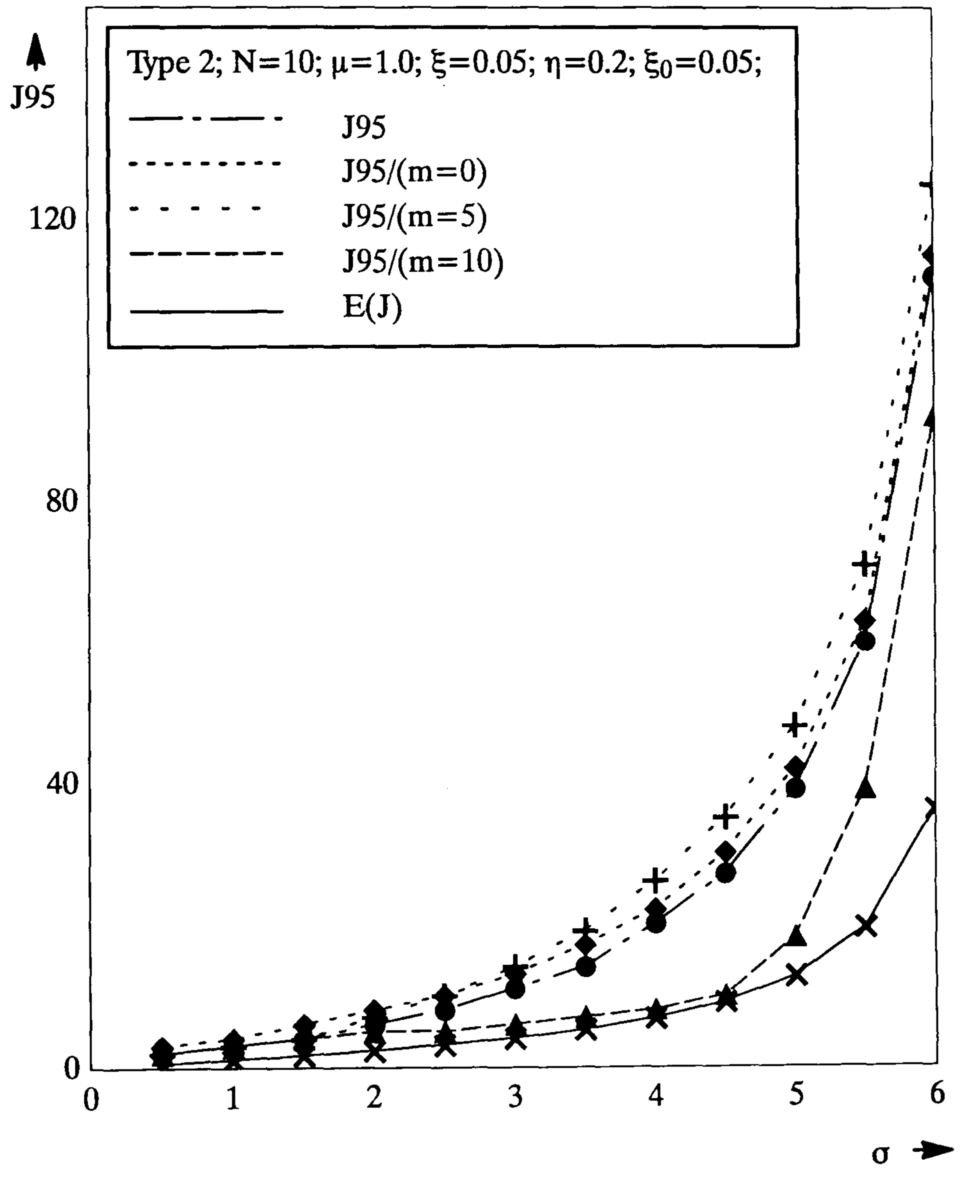

Figure 3.5: Conditional and unconditional 95-percentiles, and $E(J)$, as functions of the arrival rate, for Type 2. Service capacity=6.66667.

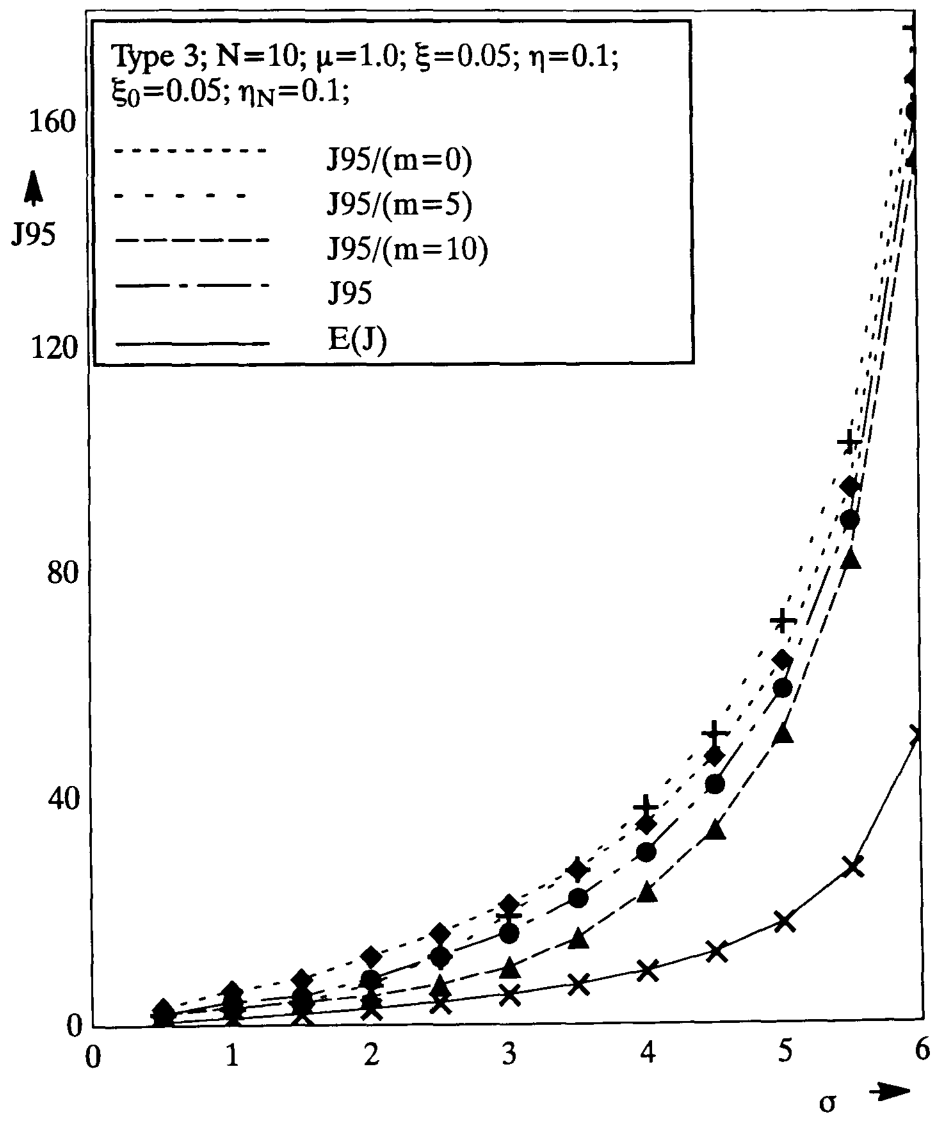

Figure 3.6: Conditional and unconditional 95-percentiles, and $E(J)$, as functions of the arrival rate, for Type 3. Service capacity=6.66667.

The unconditional percentile, $J95$, is defined by $P(J \leq J95) = 0.95$, while in the conditional percentiles the conditioning is upon the number, m, of the operative processors. The figures demonstrate that, in all these systems, the mean can be a very poor predictor of the percentiles. Also, it is interesting to observe that although Type 1 has the best average performance, it has the worst percentile conditioned upon all processors being inoperative ($m = 0$), as well as the largest spread of percentiles.

In all the above examples, the arrival rate σ is constant and independent of the operative state of the multiprocessor. Figure 3.7 deals with the case of operative state dependent arrival rates. In this example, Type 2 with 10 processors is considered. The arrival rate σ_i depends on the operative state i, where i is the number of processors in working condition. Hence, the mean arrival rate σ is given by, $\sigma = \sum_{i=0}^{N} \sigma_i * q_i$. The arrival rate during operative state i consists of two parts. One is the operarive state independent part σ_1, the other is the operative state dependent part $i\sigma_2$. The latter is taken as proportional to i. The graph is $E(J)$ vs. σ, where $\sigma = \sigma_1 + \sigma_2$. Clearly, σ_1/σ is the portion of arrival rate the is independent of the operative state. In the different graphs, this portion is different. As this portion reduces the $E(J)$ reduces. This is quite intuitive.

It is possible to work out more performance analysis and measures. Since the purpose is primarily to demonstrate the method, further analysis is not carried on.

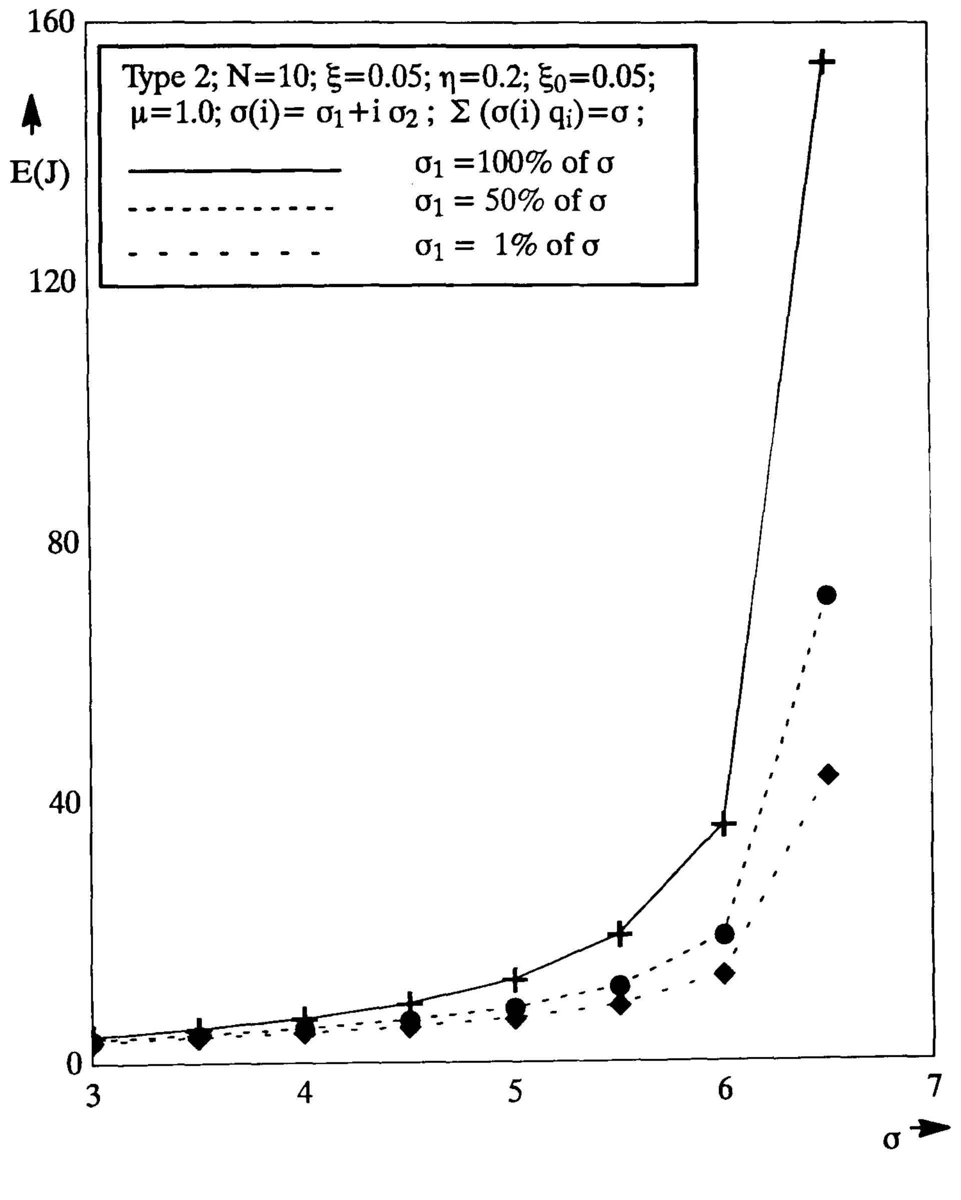

Figure 3.7: $E(J)$ vs. σ, with operative state dependent arrivals. Service capacity=6.66667.

3.3 Two servers in tandem

In this section, we take up a two-stage queueing network with a feedback, and a finite intermediate waiting room. This network, shown in Figure 3.8, is an exponential service system. Requests arrive at the system as single independent arrivals according to a Poisson process. The arrival rate is given by σ. The service rates in stages 1 and 2 are μ_1 and μ_2, respectively. Each stage has a single server, that can process only one request at any time. The principal characteristic of the service in this network is *blocking*. Stage 1 has unbounded waiting room, where as stage 2 has a finite buffer. When N requests are queued or in service in the second stage, the server in the first stage is blocked and ceases to offer service. Service in the first stage is resumed when the queue length in the second stage falls to $N - 1$. When a request is processed by the server in stage 2, it either comes out of the system, or is fed back to stage 1 for re-service, with probabilities $1 - \theta$ and θ, respectively. This network was studied by Konheim and Reiser [35]. Neuts [49] studied two-stage blocking networks, without feedback, under more general statistical hypothesis.

This system is modelled in this section, for spectral expansion solution. $I(t)$ represents the number of requests in the queue in stage 2, including the one in service. $I(t)$ can be considered as the operative state of the Markov process that represents the system. We assume this Markov process is irreducible. $J(t)$ is the number of requests queued up in stage 1, including the one being served by the server. The matrices A and A_j are given by,

$$\begin{aligned} A(i,j) &= \mu_2(1-\theta)\ ;\ i = 1,2,\ldots,N\ ;\ j = i-1\ ; \\ A(i,j) &= 0\ ,\ \textit{otherwise}\ ,\ \textit{and}\ , \\ A_j &= A\ ;\ j = 0,1,\ldots\ . \end{aligned} \qquad (3.8)$$

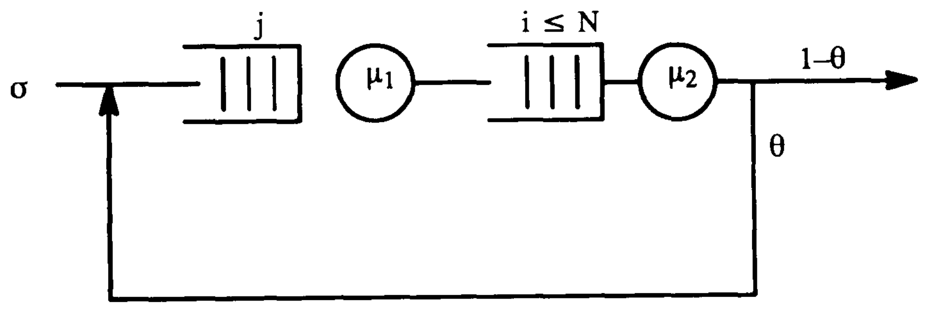

Figure 3.8: Two servers in tandem, with a feedback

A is then written as,

$$A = A_j \ (j = 0, 1, \ldots) = \begin{bmatrix} 0 & & & & \\ \mu_2(1-\theta) & 0 & & & \\ & \mu_2(1-\theta) & 0 & & \\ & & \ddots & \ddots & \\ & & & \mu_2(1-\theta) & 0 \end{bmatrix} . \tag{3.9}$$

One-step upward transitions occur (i) when a new request arrives at stage 1, or, (ii) when a request after having processed by stage 2 is fed back to stage 1 for re-service. In (i), there is no change in the operative state of the process, whereas, in (ii) a change in the operative state occurs along with the one-step upward transition. The matrices B and B_j can then be written as,

$$B = B_j \ (j = 0, 1, \ldots) = \begin{bmatrix} \sigma & & & & \\ \mu_2\theta & \sigma & & & \\ & \mu_2\theta & \sigma & & \\ & & \ddots & \ddots & \\ & & & \mu_2\theta & \sigma \end{bmatrix} . \tag{3.10}$$

The one-step downward transitions happen when a request goes from stage 1 to stage 2, after its service in stage 1. The C and C_j matrices are,

$$C = C_j \ (j = 1, 2, \ldots) = \begin{bmatrix} 0 & \mu_1 & & & \\ & 0 & \mu_1 & & \\ & & 0 & \ddots & \\ & & & \ddots & \mu_1 \\ & & & & 0 \end{bmatrix} , \tag{3.11}$$

with $C_0 = 0$. Clearly, the threshold, $M = 1$.

3.3.1 Numerical results

Two example networks that fall in this framework are considered. In the first example, the first-stage server is much faster than the second-stage server. This would need a large buffer space for good utilization. The values we have chosen are: $\mu_1 = 50.0$, $\mu_2 = 2.0$, $\theta = 0$. Two different values for the buffer capacity are considered, $N = 50, 100$. The steady state performance measures computed are, the expected number of jobs waiting for service ($E(J)$) at stage 1, expected number of jobs in the buffer ($E(I)$) waiting for service at stage 2 and the blocking probability. Blocking probability, P_b, is the steady state joint probability that the buffer at stage 2 is full and the number of jobs waiting for stage 1 service is non-zero.

Figure 3.9 shows the plot, $E(J)$ vs. σ, for the cases with $N = 50$ and $N = 100$. Figure 3.10 shows the plot $E(I)$ vs. σ for the same cases. And, figure 3.11 shows the blocking probability vs. σ.

The second example is the same as the example considered by Konheim and Reiser [35]. Here, the parameter values are, $\mu_1 = 2.0$, $\mu_2 = 1.1$, $\theta = 0$. Three different job arrival rates are considered, $\sigma = 0.5, 1.0, 1.05$. The same performance measures, as above, are computed for different values of N. These results are shown in Figures 3.12, 3.13 and 3.14.

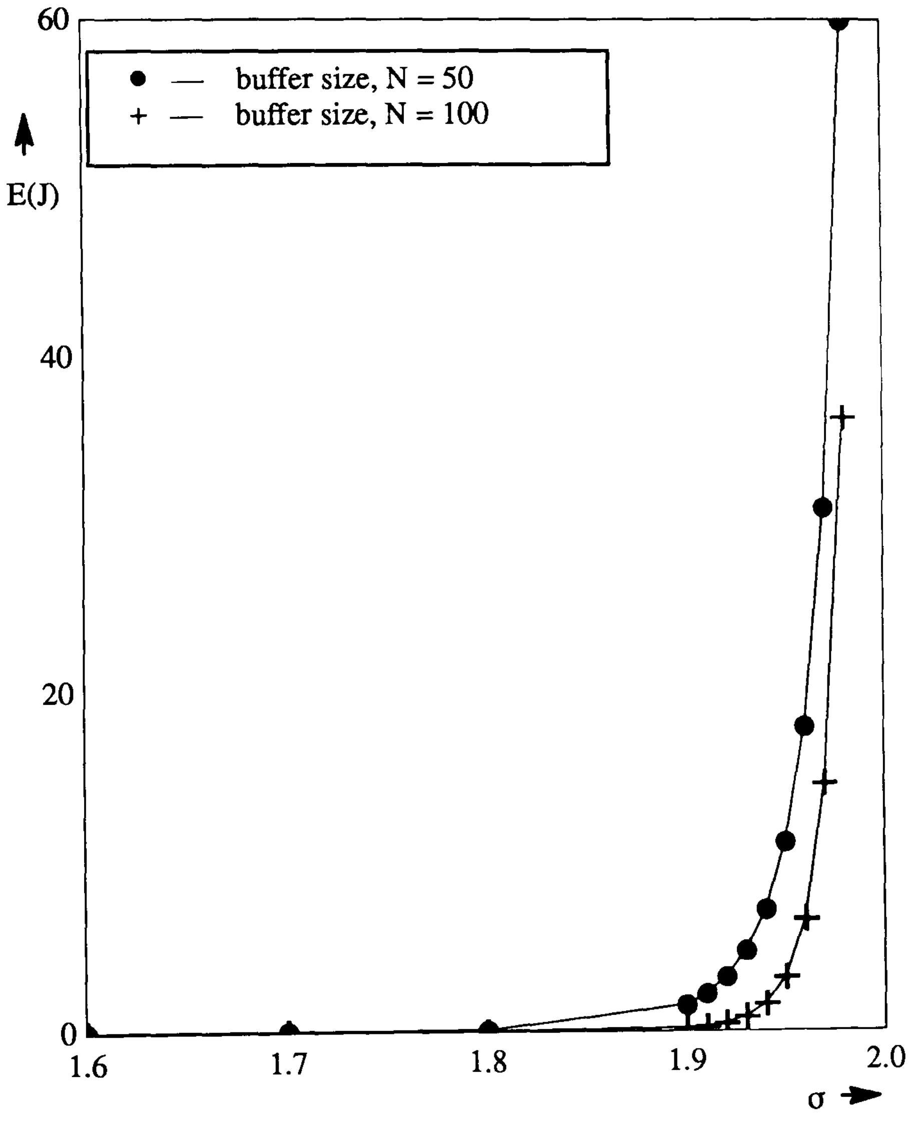

Figure 3.9: $E(J)$ vs. σ for the example with $\mu_1 = 50$, $\mu_2 = 2$, $\theta = 0$.

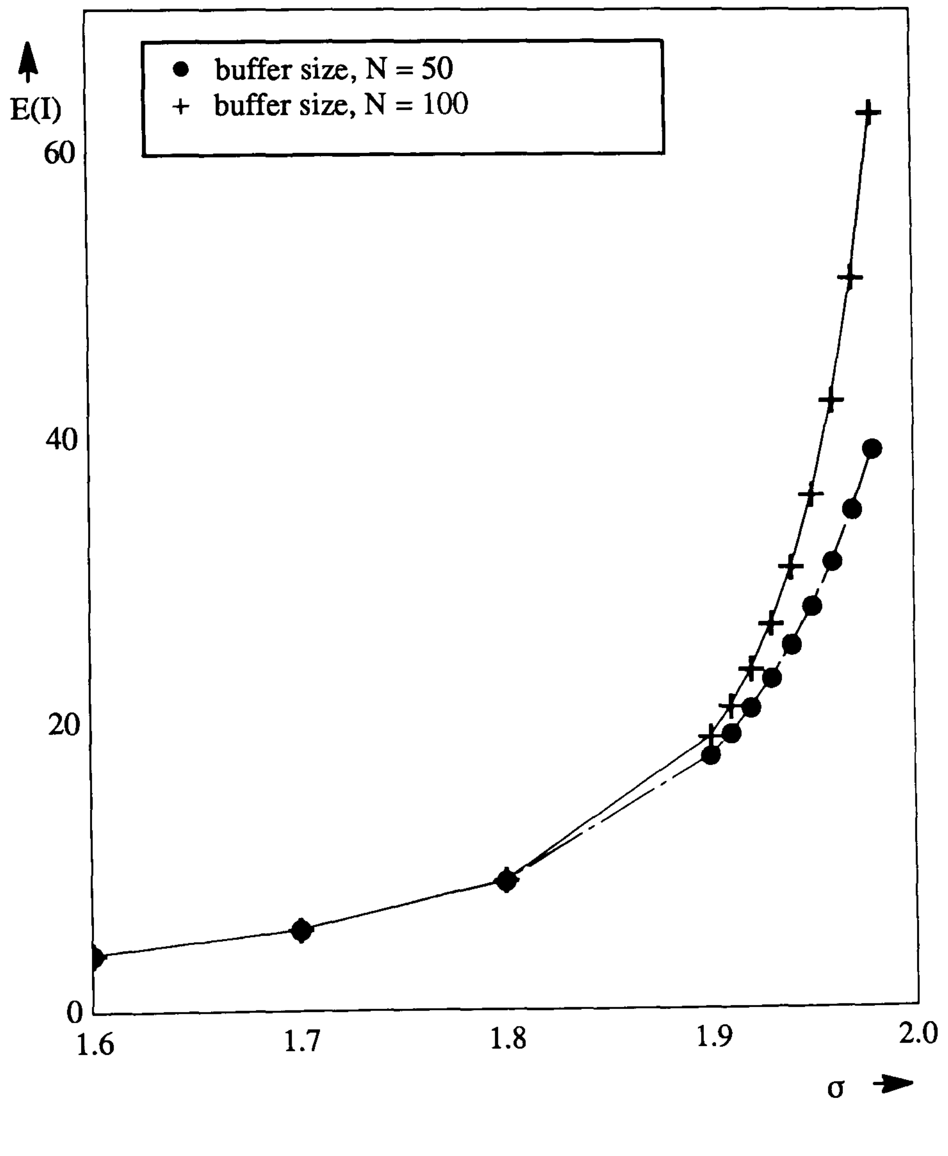

Figure 3.10: $E(I)$ vs. σ for the example with $\mu_1 = 50$, $\mu_2 = 2$, $\theta = 0$.

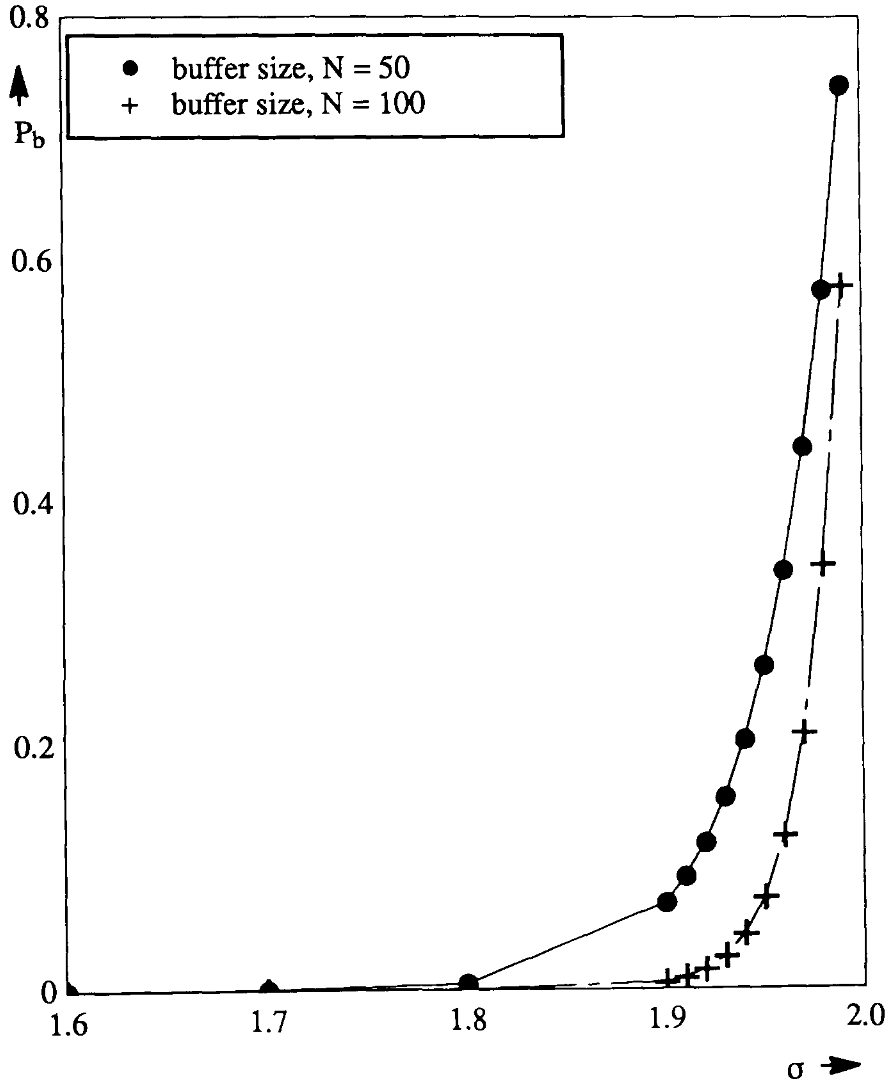

Figure 3.11: Blocking probability vs. σ for the example with $\mu_1 = 50$, $\mu_2 = 2$, $\theta = 0$.

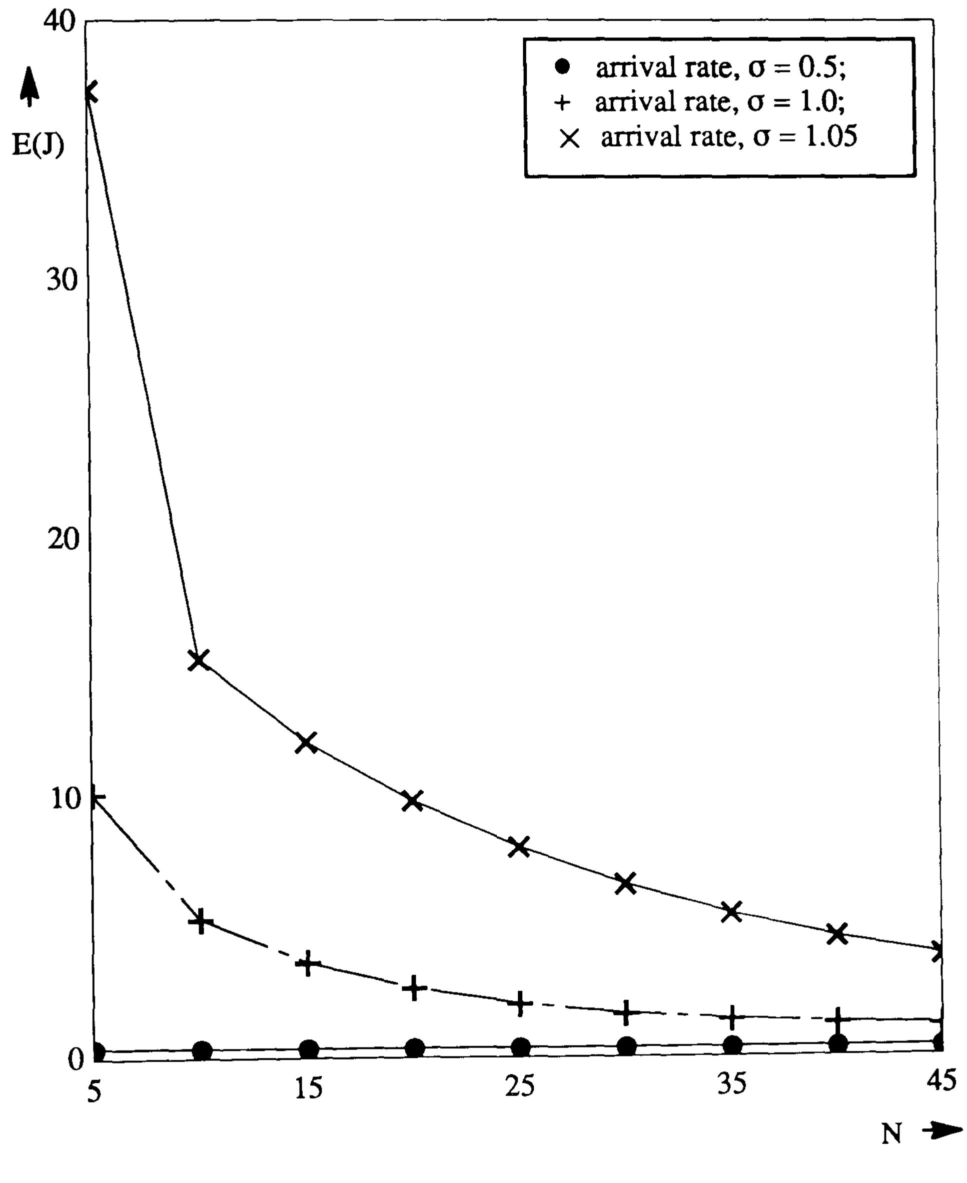

Figure 3.12: $E(J)$ vs. σ for the example with $\mu_1 = 2.0$, $\mu_2 = 1.1$, $\theta = 0$.

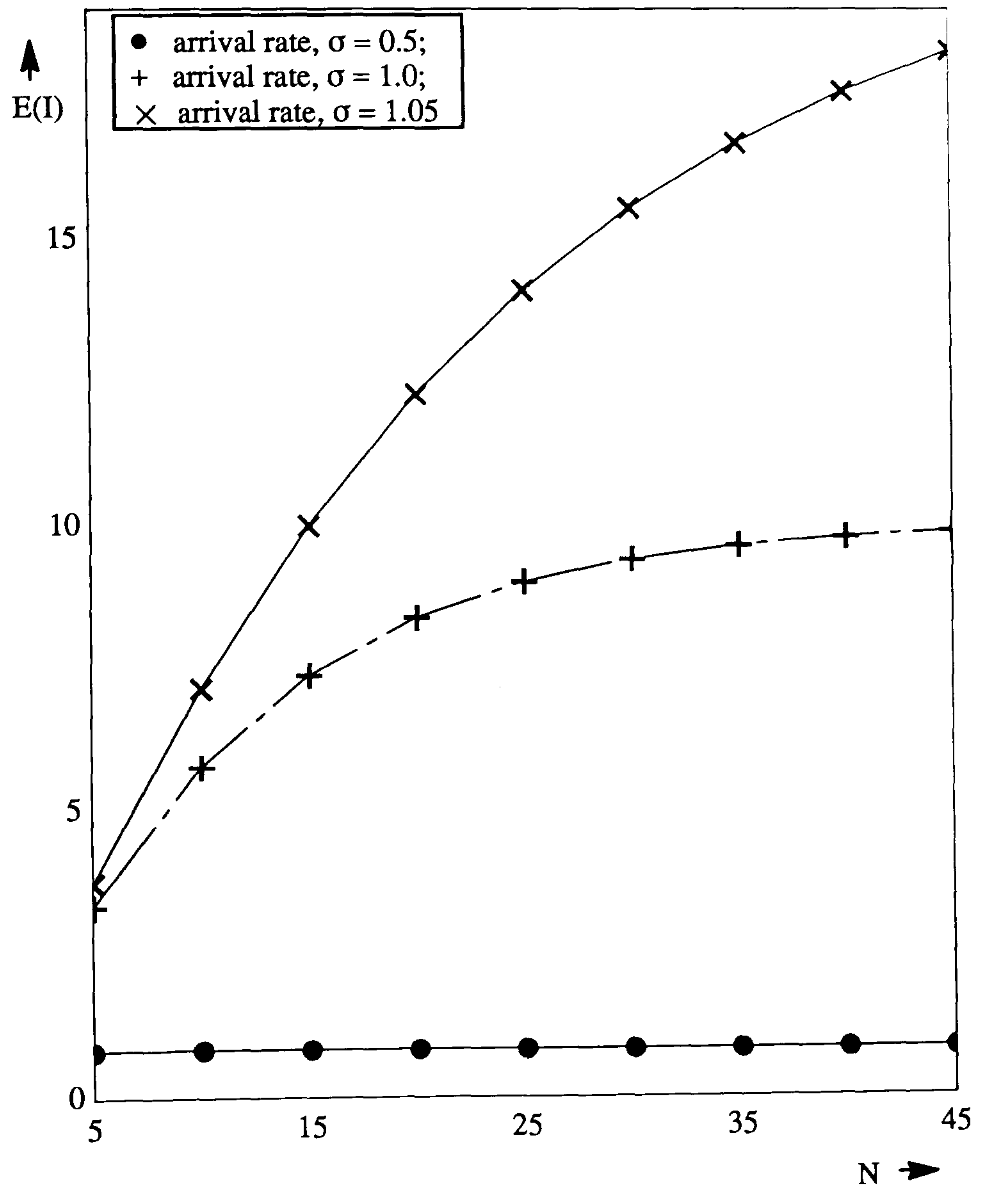

Figure 3.13: $E(I)$ vs. σ for the example with $\mu_1 = 2.0$, $\mu_2 = 1.1$, $\theta = 0$.

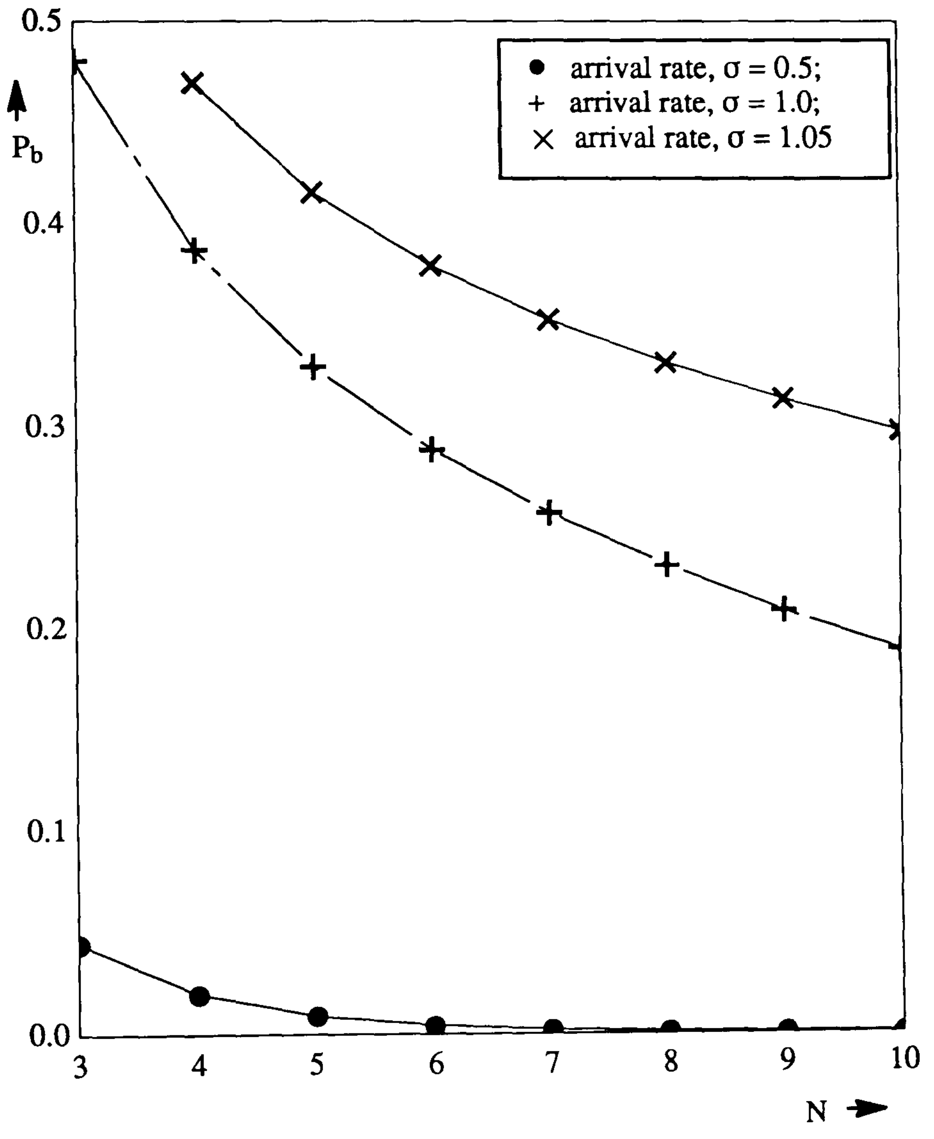

Figure 3.14: Blocking probability vs. σ for the example with $\mu_1 = 2.0$, $\mu_2 = 1.1$, $\theta = 0$.

3.4 Conclusions

This Chapter is intended to demonstrate the use of spectral expansion solution discussed in Chapter 2. This is accomplished by taking up two non-trivial examples. The first one is a homogeneous multiprocessor with general failures and repairs of processors, serving a Poisson stream of jobs. Several types of failure and repair environments are considered. These systems are evaluated for performance and dependability. Some of the results are plotted in figures, though more results can be generated. However, the model can be generalised, by allowing more complex operative states for the multiprocessor, to include arrivals, services, failures and repairs, all these of Phase type and operative state dependent.

The other example is the two stage network considered by Konheim and Reiser [35]. This network is evaluated for several sets of parameter values. The numerical performance results are shown in the figures.

More complicated applications of spectral expansion are dealt with, successfully in the following Chapters. There are many other applications and examples solved by other authors, some of them can be found in [13, 14]. Looking at the very distinct problems solved by spectral expansion, it can be said that the method is quite general and has a wide range of applicability.

3.5 Contributions

The contributions of this chapter are concerned with the application of spectral expansion method to two non-trivial problems. They are,

- A homogeneous multiprocessor system serving a stream of jobs, with unbounded queueing capacity is analysed for steady state performance and dependability. This is done in the presence of independent, as well as correlated failures and repairs of the processors. Several types of systems are considered and some of their performance characteristics are plotted.

- The other example is the two stage network considered by Konheim and Reiser [35]. This network is evaluated for several sets of parameter values.

The work on homogeneous multiprocessor system, has appeared in our paper [5].

Chapter 4

Spectral Expansion for Markov Processes of Finite State Space

4.1 Introduction

The spectral expansion methodology for the steady state solution of ergodic, irreducible Markov processes of unbounded state space, has been the subject in Chapter 2. Stability and eigen analysis of those Markov processes has also been carried out. Chapter 3 has presented some illustrative applications on how to apply those solution techniques. As has been said earlier, several extensions of spectral expansion are possible. One such extension is the subject of this Chapter.

The Markov models of many practical systems have a *finite* state space, rather than infinite one. For example, if the homogeneous multiprocessor system has a finite buffer where the incoming jobs queue up, and if the incoming jobs are rejected or lost when this buffer is full, then what results is a Markov process of a finite state space. More practical systems are of this type, compared to the types covered in Chapters 2 and 3. Hence, it is important to find efficient solution techniques to such problems, if they exist. The following section describes the Markov model of such processes. In section 4.3 and 4.4 the steady state solution and computational aspects of such a model are developed. Section 4.5 briefly describes two alternate methods with some comparative remarks. We illustrate the application of this solution by an example system, this time a finite capacity homogeneous multiprocessor system with breakdowns and repairs. This example is presented in section 4.6.

4.2 The Markov model

Let us recall the Quasi-Birth-and-Death process X described in Chapter 2. The random variable $J(t)$ of X is unbounded. Let this process be modified by imposing a limit on the maximum value of $J(t)$, and let that limit be L. This results in a Markov process Z, $Z = \{[I(t), J(t)]\,;\, t \geq 0\}$, of finite state space. We assume Z is irreducible, with a state space $\{0,1,\ldots,N\} \times \{0,1,\ldots,L\}$. This can be represented on a finite rectangular lattice strip, with $I(t)$ varying in the lateral or horizontal direction and $J(t)$ in the vertical direction. The possible transitions in this Markov process are:

(a) A_j – purely lateral transitions – from state (i,j) to state (k,j), $(0 \leq i,k \leq N\,;\, i \neq k\,;\, j = 0,1,\ldots,L)$;

(b) B_j – one-step upward transitions – from state (i,j) to state $(k,j+1)$, $(0 \leq i,k \leq N\,;\, j = 0,1,\ldots,L-1)$;

(c) C_j – one-step downward transitions – from state (i,j) to state $(k,j-1)$, $(0 \leq i,k \leq N\,;\, j = 1,2,\ldots,L)$;

Here too, the possible change in J in any transition is either $+1$ or 0 or -1. A_j, B_j and C_j are again the transition rate matrices, square matrices each of size $(N+1) \times (N+1)$, associated with (a), (b) and (c) respectively. Thus, $A_j(i,k)$, $(i \neq k)$ is the transition rate from state (i,j) to state (k,j), and $A_j(i,i) = 0$. $B_j(i,k)$ is the transition rate from (i,j) to $(k,j+1)$. $C_j(i,k)$ is the transition rate from (i,j) to $(k,j-1)$, and $C_0 = 0$, by definition. We assume the process has a threshold, an integer M, $(M \geq 1)$ such that the instantaneous transition rates of (a), (b) and (c) do not depend on j when $L \geq j \geq M$ in the case of (a), $L-1 \geq j \geq M-1$ in the case of (b), and when $L \geq j \geq M$ for (c). Thus, we get

$$\begin{aligned} A_j &= A; \quad L \geq j \geq M, \\ B_j &= B; \quad L-1 \geq j \geq M-1, \\ C_j &= C; \quad L \geq j \geq M . \end{aligned} \tag{4.1}$$

Clearly, if $L \to \infty$, the process Z becomes X. For notational convenience, we call X, the *parent* process of Z. In Chapter 2, it was explained that X is stable or ergodic if, and only if certain conditions (2.21) and (2.23) are satisfied. Whereas, these conditions are not necessary for Z to be stable. If Z is irreducible, Z is stable or ergodic. We need to find the steady state probability $p_{i,j}$ of the state (i,j) in terms of the known parameters of the system. $p_{i,j}$ is defined as:

$$p_{i,j} = \lim_{t\to\infty} P(I(t)=i, J(t)=j) \; ; \; i=0,1,\ldots,N \; ; \; j=0,1,\ldots,L. \tag{4.2}$$

D_j^A, D_j^B and D_j^C are diagonal matrices defined just as in Chapter 2, but this time for $j=0,1,\ldots,L$. D^A, D^B and D^C are the ones corresponding to the j-independent matrices A, B and C respectively, as in Chapter 2.

4.3 The solution

The vector $\mathbf{v}_j$ is again as defined in Chapter 2, but here, for $j=0,1,\ldots,L$. The steady state balance equations, in terms of $\mathbf{v}_j$, can now be written. They are:

$$\mathbf{v}_0[D_0^A + D_0^B] = \mathbf{v}_0 A_0 + \mathbf{v}_1 C_1 \; , \tag{4.3}$$

$$\mathbf{v}_j[D_j^A + D_j^B + D_j^C] = \mathbf{v}_{j-1}B_{j-1} + \mathbf{v}_j A_j + \mathbf{v}_{j+1}C_{j+1} \; ; \; j=1,2,\ldots,M-1 \; , \tag{4.4}$$

$$\mathbf{v}_j[D^A + D^B + D^C] = \mathbf{v}_{j-1}B + \mathbf{v}_jA + \mathbf{v}_{j+1}C \; ; \; j = M, M+1, \ldots, L-1 \; , \quad (4.5)$$

$$\mathbf{v}_L[D^A + D^C] = \mathbf{v}_{L-1}B + \mathbf{v}_LA \; . \quad (4.6)$$

The above are the steady state balance equations, totally $L + 1$ in the vector form, i.e. in $\mathbf{v}_j$'s, and $(L+1)*(N+1)$ in scalar form, i.e. in $p_{i,j}$'s. Of them, only $(L+1)*(N+1)-1$ are linearly independent. This is because, the generator matrix of Z is singular. For the solution of the probability distribution $\{p_{i,j}\}$, we also have a normalizing equation, given as,

$$\sum_{j=0}^{L} \mathbf{v}_j\mathbf{e} = 1.0 \; . \quad (4.7)$$

The balance equations (4.5), with j-independent coefficients, are finite in number, equal to $L-M$ in the vector form in terms of $\mathbf{v}_j$'s, and $(L-M)*(N+1)$ if written in the scalar probability form, i.e. in terms of $p_{i,j}$'s. Note, the j-independent balance equations of the process X in Chapter 2 are infinite. If L is large, then these equations (4.5) would be large in number, though finite. Spectral expansion solution for the finite Markov processes is essentially the solution to these j-independent balance equations. These equations can be rewritten as,

$$\mathbf{v}_jQ_0 + \mathbf{v}_{j+1}Q_1 + \mathbf{v}_{j+2}Q_2 = 0 \; ; \; j = M-1, M, \ldots, L-2 \; , \quad (4.8)$$

where, Q_0, Q_1 and Q_2 are as defined in Chapter 2.

Now, let us write the charactestic matrix polynomials of the processes X and $\overline{X}$, as given in Chapter 2. They are, respectvely,

$$Q(\lambda) = Q_0 + Q_1\lambda + Q_2\lambda^2 \; , \quad (4.9)$$

$$\overline{Q}(\beta) = Q_2 + Q_1\beta + Q_0\beta^2 \; . \quad (4.10)$$

The solution of (4.5), and hence the steady state solution of Z, is related to the eigenvalue-eigenvectors of $Q(\lambda)$ and those of $\overline{Q}(\beta)$. As we have seen in Chapter 2, the non-zero eigenvalues of $Q(\lambda)$ are the reciprocals of the non-zero eigenvalues of $\overline{Q}(\beta)$, with the same corresponding eigenvectors. Let (λ, ψ) be any eigenvalue-eigenvector pair of $Q(\lambda)$, and (β, ϕ) be that of $\overline{Q}(\beta)$. Then, we have,

$$\psi Q(\lambda) = 0 \; ; \; det[Q(\lambda)] = 0 \; , \tag{4.11}$$

$$\phi \overline{Q}(\beta) = 0 \; ; \; det[\overline{Q}(\beta)] = 0 \; . \tag{4.12}$$

We have analysed at length in Chapter 2, the nature of the eigenvalues and eigenvectors of both $Q(\lambda)$ and $\overline{Q}(\beta)$. These analyses hold good here too. The number of eigenvalues is related to the ranks of matrices Q_0 and Q_2. Let the rank of Q_0 be $N + 1 - r_0$, and the of Q_2 be $N + 1 - r_2$, as in Chapter 2. We have seen earlier, the number of eigenvalues of $Q(\lambda)$ is, $d = 2N + 2 - r_2$. If $Q(\lambda)$ has an eigenvalue of multiplicity n, this is counted as n eigenvalues. Also, we assume, such a multiple eigenvalue has n independent eigenvectors. We have found that so in practice. The number of zero-eigenvalues of $\overline{Q}(\beta)$, is then r_2. It can be easily seen, following similar analysis as in Chapter 2, for any eigenvalue-eigenvector pair, (λ_k, ψ_k), of $Q(\lambda)$, $\mathbf{v}_j = \psi_k \lambda_k^{j-M+1}$, for $j = M-1, M, \ldots, L$, satisfies equations (4.5). This is, then, a *particular* solution. Hence, for all the d eigenvalue and eigenvector pairs of $Q(\lambda)$, we have d particular solutions of (4.5).

Let $(\beta_k = 0, \phi_k)$ be a zero-eigenvalue and corresponding eigenvector pair of $\overline{Q}(\beta)$. Then, $\mathbf{v}_j = \phi_k \beta_k^{L-j}$, for $j = M - 1, M, \ldots, L$, also satisfies (4.5). Hence, this is a particular solution. There would be $r_2 = 2N + 2 - d$ number of particular solutions of this type, since the number of zero-eigenvalues of $\overline{Q}(\beta)$ is, $r_2 = 2N + 2 - d$.

Consider $(\beta_k \neq 0, \phi_k)$, a non-zero eigenvalue and corresponding eigenvector pair of $\overline{Q}(\beta)$. Then, there exists a non-zero eigenvalue and corresponding eigenvector pair, (λ_k, ψ_k), of $Q(\lambda)$ such that $\lambda_k = \frac{1}{\beta_k}$ and $\psi_k = \phi_k$. Then, $\mathbf{v}_j = \phi_k \beta_k^{L-j}$ is obviously a particular solution of equations (4.5), because it satisfies those equations. But, this solution is redundant, since this is already reflected or appeared in the particular solution given by $\mathbf{v}_j = \psi_k \lambda_k^{j-M+1}$, that uses (λ_k, ψ_k). Hence, the particular solutions using non-zero eigenvalues of $\overline{Q}(\beta)$ need not be considered for the general solution, if all the non-zero eigenvalues of $Q(\lambda)$ are considered.

Therefore, the general solution should involve, every zero-eigenvalue of $Q(\lambda)$ along with its corresponding eigenvector, every zero-eigenvalue of $\overline{Q}(\beta)$ along with its corresponding eigenvector, and every non-zero eigenvalue of $Q(\lambda)$ or its reciprocal, the eigenvalue of $\overline{Q}(\beta)$, and the corresponding eigenvector.

The general solution can now be given by,

$$\mathbf{v}_j = \sum_{k=0}^{d-1} a_k \psi_k \lambda_k^{j-M+1} + \sum_{k=0}^{r_2-1} b_k \phi_{z,k} \beta_{z,k}^{L-j} \; ; \; j = M-1, M, \ldots, L. \tag{4.13}$$

where, (λ_k, ψ_k), $k = 0, 1, \ldots, d-1$, are all the eigenvalue-eigenvector pairs of $Q(\lambda)$, and, $(\beta_{z,k}, \phi_{z,k})$, $k = 0, 1, \ldots, r_2 - 1$, are all the zero-eigenvalue and corresponding eigenvector pairs of $\overline{Q}(\beta)$. a_k, $k = 0, 1, \ldots, d-1$, and b_k, $k = 0, 1, \ldots, r_2 - 1$, are arbitrary constants to be determined, so as to satisfy all the remaining balance equations and the normalizing equation. Obviously, if $r_2 = 0$, then the second summation term in (4.13) vanishes.

For computational convenience, this general solution can be modified further as follows. We have seen in Chapter 2, if X is ergodic, then $\overline{X}$ is unstable. In such a case, $Q(\lambda)$ has $N+1$ eigenvalues with absolute values less than 1.0, and $\overline{Q}(\beta)$ has $N+1$ eigenvalues with absolute values less than or equal to 1.0 (one of them equal to 1.0).

On the other hand, if $\overline{X}$ is ergodic, then X is unstable. In this case, $Q(\lambda)$ has $N+1$ eigenvalues with absolute values less than or equal to 1.0 (one of them equal to 1.0), and $\overline{Q}(\beta)$ has $N+1$ eigenvalues with absolute values less than 1.0.

It is possible that both X and $\overline{X}$ are unstable. For example, that happens if $(A - D^A + B - D^B + C - D^C)$ has a rank equal to $N-1$. In this case, each of the matrix polynomials, $Q(\lambda)$ and $\overline{Q}(\beta)$, has N eigenvalues inside the unit circle and two eigenvalues equal to 1.0. The eigenvectors corresponding to these two unity eigenvalues are different, though they are the same for both the matrix polynomials.

The numerical properties of the solution would be better if care is taken to avoid large numbers resulting from the positive powers of eigenvalues greater than 1.0. Making such changes, the general solution is modified as,

$$\mathbf{v}_j = \sum_{k=0}^{N} a_k \psi_k \lambda_k^{j-M+1} + \sum_{k=0}^{N} b_k \phi_k \beta_k^{L-j} \; ; \; j = M-1, M, \ldots, L. \tag{4.14}$$

Here, λ_k, $k = 0, 1, \ldots, N$, are the $N+1$ eigenvalues, of least absolute values, of $Q(\lambda)$. These λ's, usually, are all the eigenvalues within the unit circle, and in addition, may or may not include the one on the unit circle. And, β_k, $k = 0, 1, \ldots, N$, are the $N+1$ eigenvalues, of least absolute values, of $\overline{Q}(\beta)$. These β's, usually, are, all the eigenvalues within the unit circle, and may not or may include the one on the unit circle. If $Q(\lambda)$ and $\overline{Q}(\beta)$ have two eigenvalues equal to unity, then one of these eigenvalues appears as a λ and the other appears as a β.

The vectors $\mathbf{v}_j$, $j = M-1, M, \ldots, L$, are known if a_k's and b_k's are known. The unknown arbitrary constants a_k's and b_k's are, $2N+2$ in total. Also, the other unknowns are, the vectors $\mathbf{v}_j$, $j = 0, 1, \ldots, M-2$. These $M-2$ vectors consist of $(M-1)*(N+1)$ unknown scalar probabilities. Hence, the total number of scalar unknowns are $(M+1)*(N+1)$. To get these unknowns, we have equations (4.3, 4.4, 4.6) and the normalizing equation (4.7). These are $(M+1)*(N+1)+1$ scalar, linear, simultaneous equations of which $(M+1)*(N+1)$ are independent. Hence, a unique solution exists for the unknowns.

4.4 Computation

The computational aspects are quite similar to those described in Chapter 2. First, all the eigenvalues and eigenvectors of the matrix polynomial $Q(\lambda)$ are obtained, by following the linearisation procedure given in Chapter 2. The λ's that are greater than or equal to 1.0 in absolute value are inverted to get the required non-zero β's of $\overline{Q}(\beta)$. The zero-eigenvalues and corresponding eigenvectors of $\overline{Q}(\beta)$ are just the zero-eigenvalues and corresponding eigenvectors of the matrix C. These are obtained easily, if required, without resorting to quadratic eigenvalue problem.

Computing $(M+1)*(N+1)$ unknowns from $(M+1)*(N+1)$ equations can be time consuming. Just as in Chapter 2, simplification is possible in computation. From (4.14), $\mathbf{v}_M$ and $\mathbf{v}_{M-1}$ can be expressed linearly in terms of a_k's and b_k's, using the values of the computed eigenvalues and eigenvectors. If B_j's $(j = M-2, M-3, \ldots, 0)$ are non-singular, then it is possible to express the vectors $\mathbf{v}_{M-2}, \mathbf{v}_{M-3}, \ldots, \mathbf{v}_0$, one by one, linearly in terms of a_k's and b_k's, using the equations (4.4) for $(j = M-1, M-2, \ldots, 1)$. We have, thus, got all the $\mathbf{v}_j$'s expressed in terms of a_k's and b_k's. Hence, we just need to compute the $2N+2$ unknown arbitrary constants a_k, b_k $(k = 0, 1, \ldots, N)$, using the remaining equations (4.3, 4.6, 4.7). Equations (4.3, 4.6), after substituting for $\mathbf{v}_0$, $\mathbf{v}_1$, $\mathbf{v}_L$ and $\mathbf{v}_{L-1}$, are $2N+2$ linear scalar equations in a_k's and b_k's. Of them, only $2N+1$ are independent. The independent equation (4.7) can also be written in terms of a_k's and b_k's, by suitable substitution. Hence, we now have $2N+2$ independent linear equations with $2N+2$ unknowns, a_k's and b_k's.

If λ_k is real then, $\boldsymbol{\psi}_k$ and a_k would be real. If λ_k and λ_{k+1} are complex conjugate eigenvalues, then the corresponding eigenvectors, $\boldsymbol{\psi}_k$ and $\boldsymbol{\psi}_{k+1}$, and the corresponding arbitrary constants, a_k and a_{k+1}, are also complex conjugate pairs. Then, it is possible to write,

$$\begin{aligned} a_k\boldsymbol{\psi}_k\lambda_k^{j-M+1} + a_{k+1}\boldsymbol{\psi}_{k+1}\lambda_{k+1}^{j-M+1} &= 2*Re(a_k)Re(\boldsymbol{\psi}_k\lambda_k^{j-M+1}) - \\ &\quad 2*Im(a_k)Im(\boldsymbol{\psi}_k\lambda_k^{j-M+1})\,. \end{aligned}$$

Let l_1 be the number of real eigenvalues and m_1 be the number of complex eigenvalues of the $N+1$ eigenvalues of $Q(\lambda)$ that are under consideration. Also, let l_2 and m_2 be those of $\overline{Q}(\beta)$. We know, $l_1 + m_1 = l_2 + m_2 = N+1$ and m_1, m_2 are even. Following the same simplification procedure as in Chapter 2, the general solution can be modified as,

$$\mathbf{v}_j = \sum_{k=0}^{l_1-1} x_k \psi_{r,k} \lambda_{r,k}^{j-M+1} + 2 \sum_{k=0}^{m_1/2-1} [x_{l_1+2k} Re(\psi_{c,2k} \lambda_{c,2k}^{j-M+1}) - x_{l_1+2k+1} Im(\psi_{c,2k} \lambda_{c,2k}^{j-M+1})]$$
$$+ \sum_{k=0}^{l_2-1} y_k \phi_{r,k} \beta_{r,k}^{L-j} + 2 \sum_{k=0}^{m_2/2-1} [y_{l_2+2k} Re(\phi_{c,2k} \beta_{c,2k}^{L-j}) - y_{l_2+2k+1} Im(\phi_{c,2k} \beta_{c,2k}^{L-j})] \ . \tag{4.15}$$

where x_k, y_k $(k = 0, 1, \ldots, N)$ are the real arbitrary constants. As before, the suffix $\{r, k\}$ stands for real and $\{c, k\}$ for complex parameters.

With all these simplifications, it boils down to determining $2N+2$ real unknowns, x_k's and y_k's, from that many linear simultaneous equations. As can be seen, the computational requirements do not depend on L. This is the advantage of this method.

4.5 Alternate methods

There are alternate methods for this problem, i.e. the steady state solution of the process Z. Three methods are briefly explained in this section. One is Seelen's method [55], second is the matrix-geometric solution, and the third one is by Meier-Hellstern [41] using block Gauss-Seidel iteration.

4.5.1 Seelen's method

This algorithm is based on the iterative solution of linear simultaneous equations by using successive overrelaxation and aggregation. Large number of equations can be handled. All the balance equations (4.3, 4.4, 4.5, 4.6, 4.7) are taken up together, for the simultaneous solution of all the state probabilities. A dynamically adjusted relaxation factor and a simplified structure due to an adaptation of the disaggregation step are used [55]. In each iteration, each of the state probabilities is computed. Hence, the computational time requirement for each iteration may well be proportional to $L + 1$. Note that, computation time requirement in the case of spectral expansion does not depend on L. However, it is claimed that the number of iterations needed for accurate results does not very much depend on L. The convergence is not proved. There is another drawback of this method. That is, the use of appropriate value for the relaxation parameter is important to achieve computational efficiency. There is no definite method to determine the optimal value of this relaxation parameter. This method has been extensively applied to a large number of queueing models that have been tabulated [56].

4.5.2 Matrix-geometric solution

Another method that is applicable for this problem is the matrix-geometric method. The matrix-geometric solution [51, 50], for the process X is briefly stated in Chapter 5. Similar to the case of spectral expansion, the matrix-geometric solution is essentially for the equations (4.5). That solution is given by,

$$\mathbf{v}_j = \mathbf{w}_1 R_1^{j-M+1} + \mathbf{w}_2 R_2^{L-j} \ ; \ j = M-1, M, \ldots, L \ . \tag{4.16}$$

Here, $\mathbf{w}_1$ and $\mathbf{w}_2$ are real unknown vectors of size $N + 1$ each. Hence, these are $2N + 2$ real scalar unknowns. R_1 and R_2 are an appropriate solution pair of the respective quadratic matrix equations,

$$\begin{aligned} Q_0 + R_1 Q_1 + R_1^2 Q_2 &= 0 \ , \\ Q_2 + R_2 Q_1 + R_2^2 Q_0 &= 0 \ . \end{aligned} \tag{4.17}$$

It is to be noted, not all pairs of solution of equations (4.17) are suitable for the matrix-geometric solution (4.16). One solution that is appropriate is, the one in which R_1 and R_2 are minimal non-negative. An iterative matrix-geometric solution for computing R_1 and R_2 from (4.17), is explained in Chapter 5. An algorithm for obtaining of an appropriate solution of equations (4.17) is also given in the recent paper [65].

It can be seen, the expressions for $\mathbf{v}_j$ in (4.16) satisfy the equations (4.5). Once, R_1 and R_2 are known, the unknown vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ can be determined using the remaining equations (4.4, 4.5, 4.7). That procedure is quite similar to that of spectral expansion, with $2N + 2$ real unknowns and $2N + 2$ linear simultaneous equations.

However, the computation time requirements for computing R_1 and R_2 are much greater than for computing the eigenvalues and eigenvectors in the spectral expansion solution. Here too, the computational requirements do not depend on L. And, it is shown is Chapter 5, this solution technique is less efficient compared to spectral expansion.

4.5.3 Block Gauss-Seidel iteration

This method was used by Meier-Hellstern [41], for solving all the linear simultaneous equations (4.3, 4.4, 4.5, 4.6, 4.7), iteratively, in order to compute all the $\mathbf{v}_j$'s. This was done in the context of analysing a queueing submodel, in the study of overflow models in teletraffic networks. The number of operations required in each iteration is of the order, $O((L+1)(N+1)^2)$. The paper does not mention about dependence of the number iterations needed, for a specified accuracy, on L and N. Still, the computational efficiency depends on L.

4.6 Numerical examples

As an illustration of the method, just as in Chapter 3, a homogeneous multiprocessor system is considered. But here, the multiprocessor has a finite buffer and hence the job queue is bounded, unlike the example in Chapter 3. This system is represented by the process Z, and not by X. The multiprocessor has N processors, with individual independent breakdowns and repairs of the processors. In addition, 'global' breakdowns and 'global' repairs are possible. This corresponds to Type 3 multiprocessor of Chapter 3. The number of operative states are $N+1$, designated 0 to N. Operative state i means the number of processors in working condition are i.

The other parameters of this system are: L is the size of the buffer, μ is the service rate of each processor, ξ is the independent failure rate of each processor, ξ_0 is the global simultaneous failure rate of all working processors at any time, η is the independent repair rate of each of the processors, η_N is the global simultaneous repair rate of all the broken processors at any time. The arrival rate of jobs to the multiprocessor is σ. The service, failure and repair times are exponentially distributed.

Jobs that arrive are placed in the bounded queue, if the buffer is not full at their arrival epoch. On the other hand, if the buffer is full when a new job arrives, then that job is lost or discarded by the system. Hence, not all jobs are processed by the system. Notice, there is no ergodicity condition on σ, a steady state exists for all values of σ. The job service disciple is the same as in Chapter 3.

The parameters of the Markov process Z representing this system are given as follows. M is the threshold given by, $M = N$. The matrices A and A_j are as in equation (3.5), but for $j = 0, 1, \ldots, L$. The matrices B and B_j are as given by,

$$B = B_j \ (j = 0, 1, \ldots, L-1) = diag[\sigma, \sigma, \ldots, \sigma] \tag{4.18}$$

and, B_L can be assumed equal to 0. The matrices C and C_j are as given in equation (3.2), but for $j = 1, 2, \ldots, L$. $C_0 = 0$, and $C_j = C$ for $j \geq N$.

The spectral expansion solution for v_j, hence, is valid for $L \geq J \geq N-1$. Some example systems are solved and the numerical results are presented in the figures.

Figures 4.1 and 4.2 show some of the results for the multiprocessor with finite capacity. The multiprocessor has 10 processors. The other parameters are, $\mu = 1.0$, $\xi = 0.05$, $\xi_0 = 0.05$, $\eta = 0.1$, $\eta_N = 0.1$. Figure 4.1 shows the $E(J)$, expected number of jobs in the system vs. σ, the arrival rate of jobs. This is done for various values of L. Figure 4.2 is a plot between the percentage of jobs lost unserviced, P_{loss}, as a function of the buffer the capacity L, for various values of σ.

4.7 Conclusions

In this section, the spectral expansion method is successfully extended as a solution technique for certain Markov processes of finite state space. These processes are very useful in modelling frequently occuring problems in computer and communication systems [41]. The method is exact, and is preferable to many other popular methods due to better computational efficiency. In order to demonstrate this method, a finite capacity multiprocessor example is taken up. Some interesting numerical results are presented via graphs. The method is applicable, by suitable modelling, to the multiprocessor in a general Markovian environment, for example with phase type breakdown and repair distributions, and for phase type arrival processes.

4.8 Contributions

This chapter has two contributions. They are,

- Spectral expansion method has been successfully extended to Markov processes of *finite* state space. An efficient algorithm for numerical solution, is suggested. The method is exact. Some conclusions on the computational efficiency of this method, when compared with other methods, are drawn. The advantages of the method are explained.

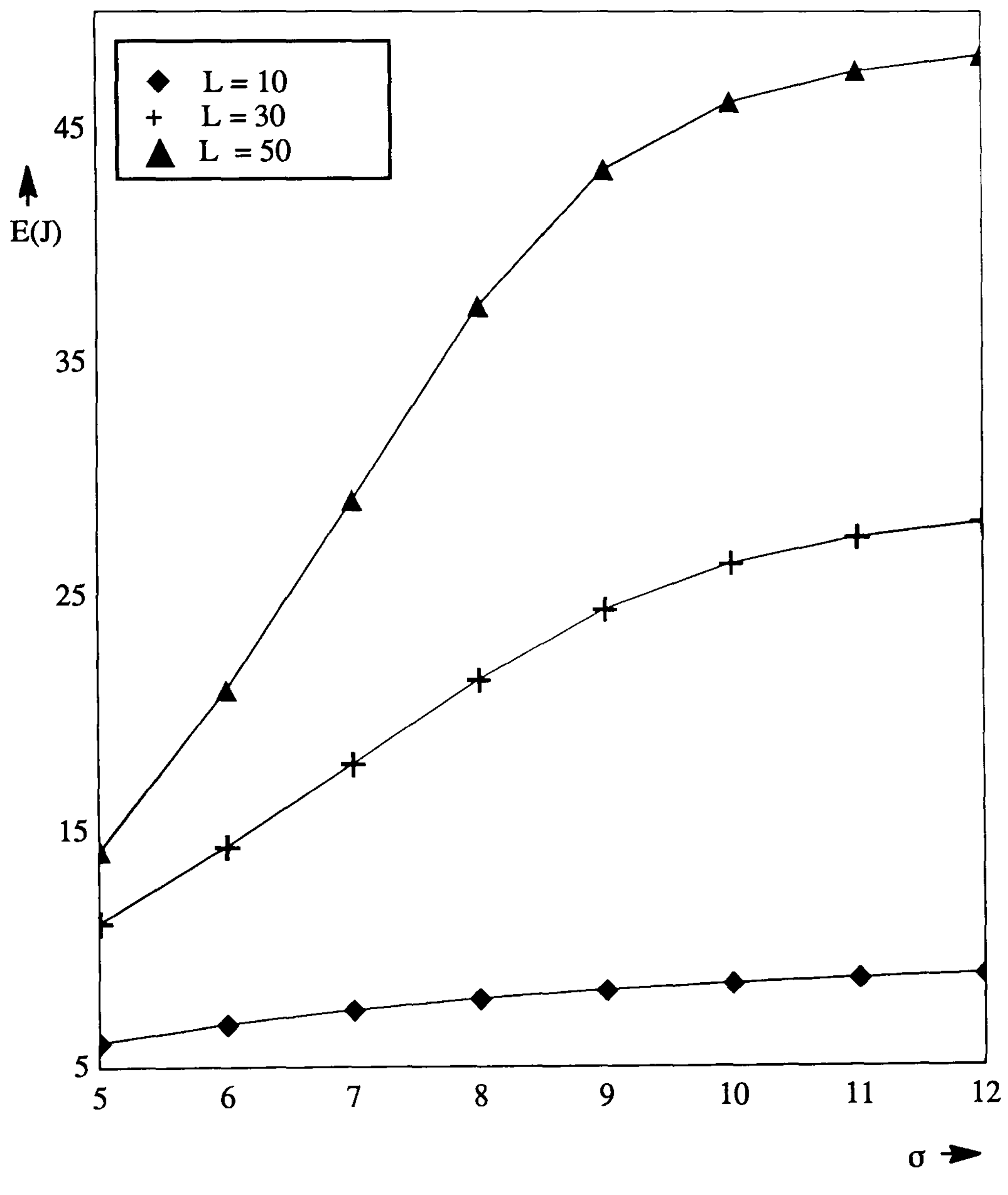

Figure 4.1: Average queue size as a function of the arrival rate.

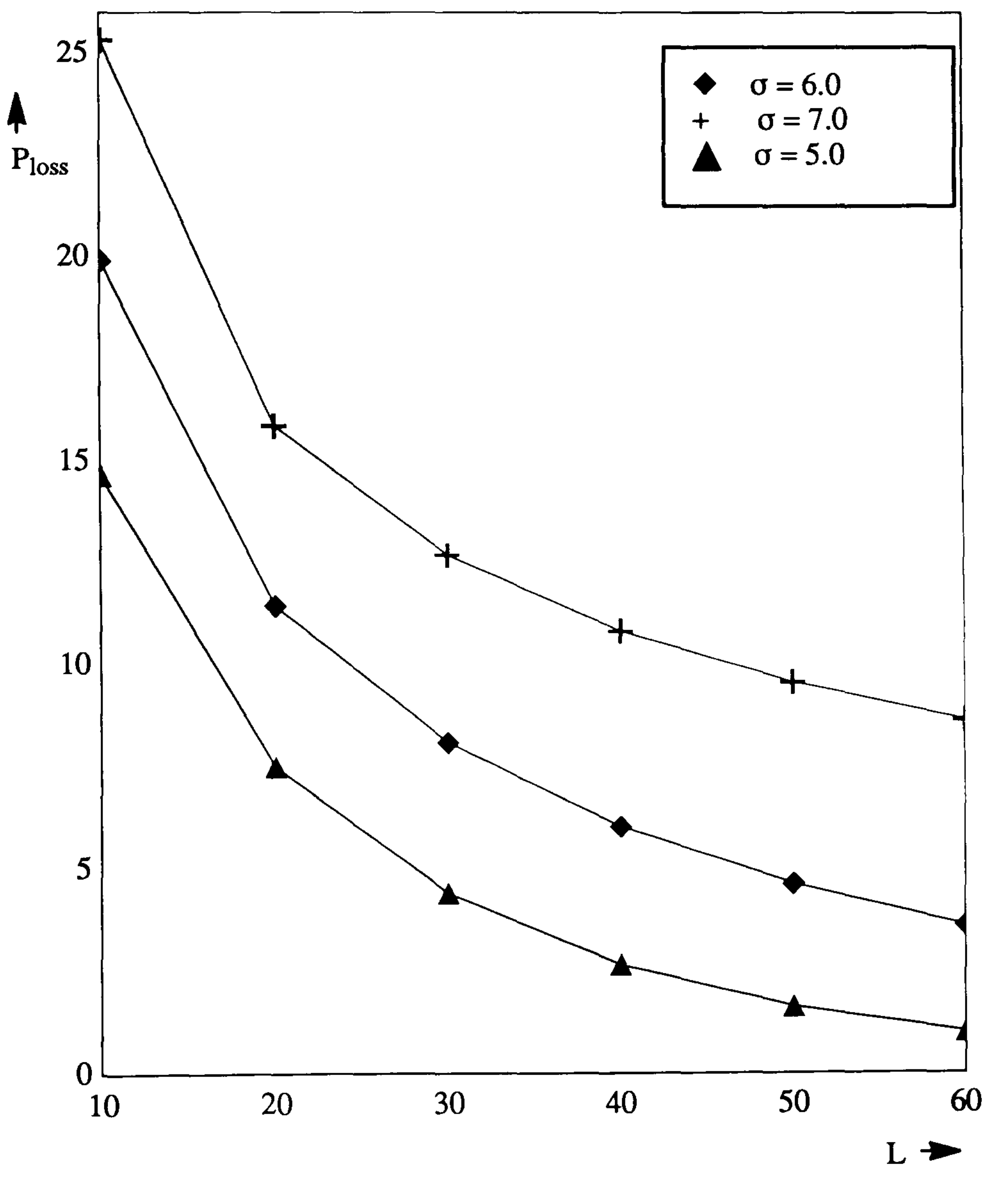

Figure 4.2: Percentage of jobs lost vs. buffer capacity.

- In order to demonstrate the application of this method, a homogeneous multiprocessor with finite queueing capacity, serving a Poisson stream of jobs, with independent or correlated server failures and repairs, is analysed for steady state performance and dependability. The numerical results are plotted.

Parts of the contributions of this chapter have been presented in our paper [9].

Chapter 5

A Comparison with the Matrix-Geometric Method

5.1 Introduction

Three Markov models, X, Y and Z, have so far been described, analysed, and solved in Chapters 2 and 4, using spectral expansion method. They have a great deal of utility in many complex modelling problems in the design and evaluation of computing systems, modern communication systems and flexible manufacturing systems. These models can also be solved by the presently most popular method, the matrix-geometric method [51, 50]. Obviously, the question of comparing these two methods, spectral expansion and the matrix-geometric method, would now arise. That is the objective of this chapter. This is done for the process X. Section 5.2 describes the matrix-geometric solution of X.

5.2 Matrix-geometric solution

The matrix-geometric method [15, 66, 51] is presently the most popular method for the solution of the Quasi-Birth-and-Death process X. Referring to Chapter 2, the balance equations remain the same, and matrix-geometric method also is essentially for the solution of the infinitely many equations (2.7). That solution [51] is given by,

$$\mathbf{v}_j = \mathbf{v}_{M-1} R^{j-M+1} \; ; \; j \geq M-1 \; , \tag{5.1}$$

where R is the *minimal non-negative solution* of the quadratic matrix equation given by,

$$Q_0 + RQ_1 + R^2 Q_2 = 0 \; . \tag{5.2}$$

Once R is known, it is possible to express $\mathbf{v}_j$'s, $j > M-1$, in terms of the unknown vector $\mathbf{v}_{M-1}$ multiplied to the right by a known matrix. This is equivalent to expressing those vectors linearly in terms of the unknown probabilities contained in $\mathbf{v}_{M-1}$.

There is a close relationship between the matrix equation (5.2) and the eigenvalue-eigenvector equation (2.11). If a matrix R satisfies (5.2), and $\psi R = \lambda \psi$ for some ψ and λ, then those ψ and λ satisfy (2.11). Conversely, any $N+1$ numbers λ_k and independent vectors ψ_k that satisfy (2.11), determine a matrix R which has them as eigenvalues and eigenvectors, and satisfies (5.2). For an ergodic system, the minimal solution of (5.2) is the one whose eigenvalues are the $N+1$ solutions of (2.11) that are inside the unit disk.

In the more general case of section 2.7, that is for the process Y, the matrix-geometric solution is still of the form (5.1), but the matrix R is obtained as the minimal non-negative solution of a matrix equation of a higher order equal to $y = y_1 + y_2$.

5.2.1 Computation

The computational steps in the matrix-geometric method are:

(1) Compute the minimal non-negative solution, R, of the quadratic matrix equation 5.2.

(2) Obtain $\mathbf{v}_M$ from the equation (5.1) in terms of the unknown vector $\mathbf{v}_{M-1}$. Now, solve the finite set of linear equations (2.6) and (2.8) for the vector $\mathbf{v}_{M-1}$ and the vectors $\mathbf{v}_j$, $j \leq M-2$. The solution is done.

(3) Use the obtained solution for the purpose of determining various moments, marginal distributions, percentiles and other system performance measures that are of interest.

There are several iterative procedures for solving equation (5.2). We have used the popular method, so-called *modified SS method* [25, 50]:

$$R_{n+1} = [-Q_0 - R_n^2 Q_2]Q_1^{-1} \quad , \tag{5.3}$$

where R_n is the computed value of R after n^{th} iteration, starting with $R_0 = 0$. The number of iterations depend on the model parameters and on the desired accuracy. The computational effort needed for this step is discussed in the following sections.

Step 2 can be simplified, as in the case of spectral expansion. That is, once $\mathbf{v}_j$'s, $j \geq M$, are expressed in terms of $\mathbf{v}_{M-1}$, it is possible to express all the remaining $\mathbf{v}_j$'s, $j \leq M-2$, in terms of $\mathbf{v}_{M-1}$ using the equations 2.6, if B_j, $j = M-2, M-3, \ldots, 0$, are non-singular. It then remains to solve for the $N+1$ scalar probabilities in $\mathbf{v}_{M-1}$ using the $N+1$ remaining scalar linear simultaneous equations. This part of the solution is very similar to step 2 of the spectral expansion procedure in section 2.6, and has the same order of complexity.

5.3 A comparison

Thus the main operational difference between the two approaches is that spectral expansion involves the computation of eigenvalues and eigenvectors of a given matrix, whereas the matrix-geometric method requires the determination of an unknown matrix R by solving a matrix equation. To evaluate the impact of that difference, we have taken up an example and performed a series of experiments where the same models were solved by both the methods. That example is the homogeneous multiprocessor system of Chapter 3. It is of Type 3, that is, with individual and global failures, and individual and global repairs. The parameters of the system that are same in all these experiments are, $\mu = 1.0$, $\xi = 0.05$, $\xi_0 = 0.05$, $\eta = 0.1$, $\eta_N = 0.1$. The performance measure that is computed, in these comparison experiments, is $E(J)$, the average number of jobs in the system. The aim of these experiments is to examine the issues of efficiency, accuracy, numerical stability and limits of application in both the solutions.

The tasks that are similar in both methods are, whenever possible, carried out by the same piece of software. For instance, the solution of a set of simultaneous linear equations is performed by routine F04ATF from the NAG numerical algorithms package [48]. That program produces diagnostic messages if the input matrix is singular or ill-conditioned. Other NAG routines are used for matrix inversion and , to find eigenvalues and eigenvectors (in the case of spectral expansion). No attempt is made, in either method, to optimize the code or to exploit particular model features. Double precision arithmetic is used throughout (except inside the NAG routines, which sometimes employ extended precision). All the programs are run on a SUN SPARC station.

It should be emphasized that both solution methods are exact in principle. That is, if the eigenvalues and eigenvectors on the one hand, and the matrix R on the other, are determined exactly, and if the respective sets of linear equations are solved exactly, then in both cases the values of the resulting probabilities and performance measures would be correct. In practice, the accuracy of the results depends on many factors and varies between the methods.

The routine that solves the linear eigenvalue-eigenvector problem proceeds as follows: (i) the matrix is transformed into upper Hessenberg form; (ii) the QR algorithm is applied to bring it into real Schur form, with 1×1 and/or 2×2 blocks on the main diagonal; (iii) the real eigenvalues (and corresponding eigenvectors) are obtained from the 1×1 blocks, and the pairs of complex-conjugate ones from the 2×2 blocks. Step (ii) involves iterations, but their number is small (an average of 2 per diagonal element). In fact, if the QR algorithm, which works to machine precision, does not complete after a total of $30 * (N+1)$ iterations, the routine quits, producing a diagnostic message. This has never happened in our experiments.

The overall computational complexity of determining all eigenvalues and eigenvectors is $O(N^3)$. Moreover, that complexity is more or less independent of the parameters of the model. Even when two eigenvalues coincide, or almost coincide, the algorithm finds linearly independent eigenvectors if the latter exist. The accuracy of the spectral expansion solution is limited only by the precision of the numerical operations. When there is no diagnostic about an ill-conditioned set of linear equations, the value of $E(J)$ computed by this method is observed to be correct to about 10 significant digits in the case of a 10-processor system. It is exact for all practical purposes. This can be seen from the following experiment.

In the 10-processor Type 3 example system, the mean queuelength $E(J)$ is computed by the spectral expansion method. Also, $E(J)$ is computed by the matrix-geometric method, for varying number of iterations. It is observed for these systems, with any number of processors, the $E(J)$ computed by the matrix-geometric method increases with the number of iterations initially, and approaches a constant value when the number of iterations become large or tend to infinity. That constant or saturated value can then be considered as the exact value of $E(J)$, ignoring the inaccuracies caused by limitations of the computed used. Table 5.1 shows the $E(J)$ computed by both the methods.

Table 5.1: Accuracy of the spectral expansion solution.

$\sigma = 6.0$;	$E(J)_{(spect.exp.)} = 50.405820557249727$
iterations	$E(J)_{(mat.geom.)}$
10	19.245439820303740
100	38.222381352764109
1000	50.404747588612197
2000	50.405820524441811
3000	50.405820557253413
4000	50.405820557253413
5000	50.405820557253413
6000	50.405820557253413
7000	50.405820557253413
8000	50.405820557253413
9000	50.405820557253413
10000	50.405820557253413
100000	50.405820557253413
1000000	50.405820557253413

After nearly 3000 iterations $E(J)_{(mat.geom.)}$ approaches a constant value and does not change upto 15 decimal points. Hence, that value $E(J) = 50.405820557253413$ can be considered exact. $E(J)_{(spect.exp.)}$ coincides with this upto the first 10 decimal points.

The matrix-geometric solution relies on the matrix R, which is computed approximately by the iterative procedure described earlier. The criterion for terminating that procedure would normally include some trade-off between computation time and accuracy. The situation is illustrated in Table 5.2, where a 10-server model, i.e. one of the examples in Chapter 3, is solved by both the methods. The iterative procedure (5.3) is terminated when

$$\max_{i,j} |R_{n+1}(i,j) - R_n(i,j)| < \epsilon ,$$

for a given value of ϵ. The performance measure that is computed is the average queue size, $E(J)$, for three different values of the arrival rate, σ. The spectral expansion solution is of course independent of ϵ. Column 4 gives the percentage relative difference $100 * [E(J)_{(spect.exp.)} - E(J)_{(mat.geom.)}]/E(J)_{(spect.exp.)}$.

The table confirms that when ϵ decreases, the matrix-geometric solution approaches the spectral expansion one. However, it is important to observe that the accuracy of R is not related in any obvious way to the accuracy of $E(J)$. Thus, taking $\epsilon = 10^{-6}$ yields an answer whose relative error is 0.0004% when $\sigma = 3$, 0.06% when $\sigma = 6$, and 6.3% when $\sigma = 6.6$. Another important aspect of the table is that, for a given ϵ, the number of iterations required to compute R increases with σ. Hence, in any of these experiments, there is no way of knowing a priori a suitable value for ϵ, to a specified accuracy requirement of the performance measures.

The computational complexity of the two solutions is compared in Figure 5.1. In both cases, the time taken to compute $E(J)$ is plotted against the arrival rate, σ. For the spectral expansion solution, that time is independent of σ and is about 0.6 seconds. The plots for the matrix-geometric solution are obtained as follows: First, the program is run with varying numbers of iterations, in order to establish the exact number of iterations which yields the value of $E(J)$ with the desired accuracy (relative to the spectral expansion result). A run with that fixed number of iterations is then timed. (In the absence of an exact result, one would have to rely on internal accuracy checks for the termination criterion.)

Table 5.2: Trade-off between accuracy and complexity.

$\sigma = 3.0$; $E(J)_{(spect.exp.)} = 5.199710420277$			
ϵ	iterations	$E(J)_{(mat.geom.)}$	% difference
10^{-3}	29	5.175072879090	0.4738252555
10^{-6}	93	5.199686527032	0.0004595110
10^{-9}	158	5.199710396767	0.0000004521
10^{-12}	223	5.199710420254	0.0000000004
$\sigma = 6.0$; $E(J)_{(spect.exp.)} = 50.4058205572$			
ϵ	iterations	$E(J)_{(mat.geom.)}$	% difference
10^{-3}	77	35.0382555318	30.4876795091
10^{-6}	670	50.3726795846	0.0657483051
10^{-9}	1334	50.4057872349	0.0000661080
10^{-12}	1999	50.4058205241	0.0000000657
$\sigma = 6.6$; $E(J)_{(spect.exp.)} = 540.46702456$			
ϵ	iterations	$E(J)_{(mat.geom.)}$	% difference
10^{-3}	77	58.19477818	89.23250160
10^{-6}	2636	506.34712584	6.31303986
10^{-9}	9174	540.42821366	0.00718099
10^{-12}	15836	540.46698572	0.00000719

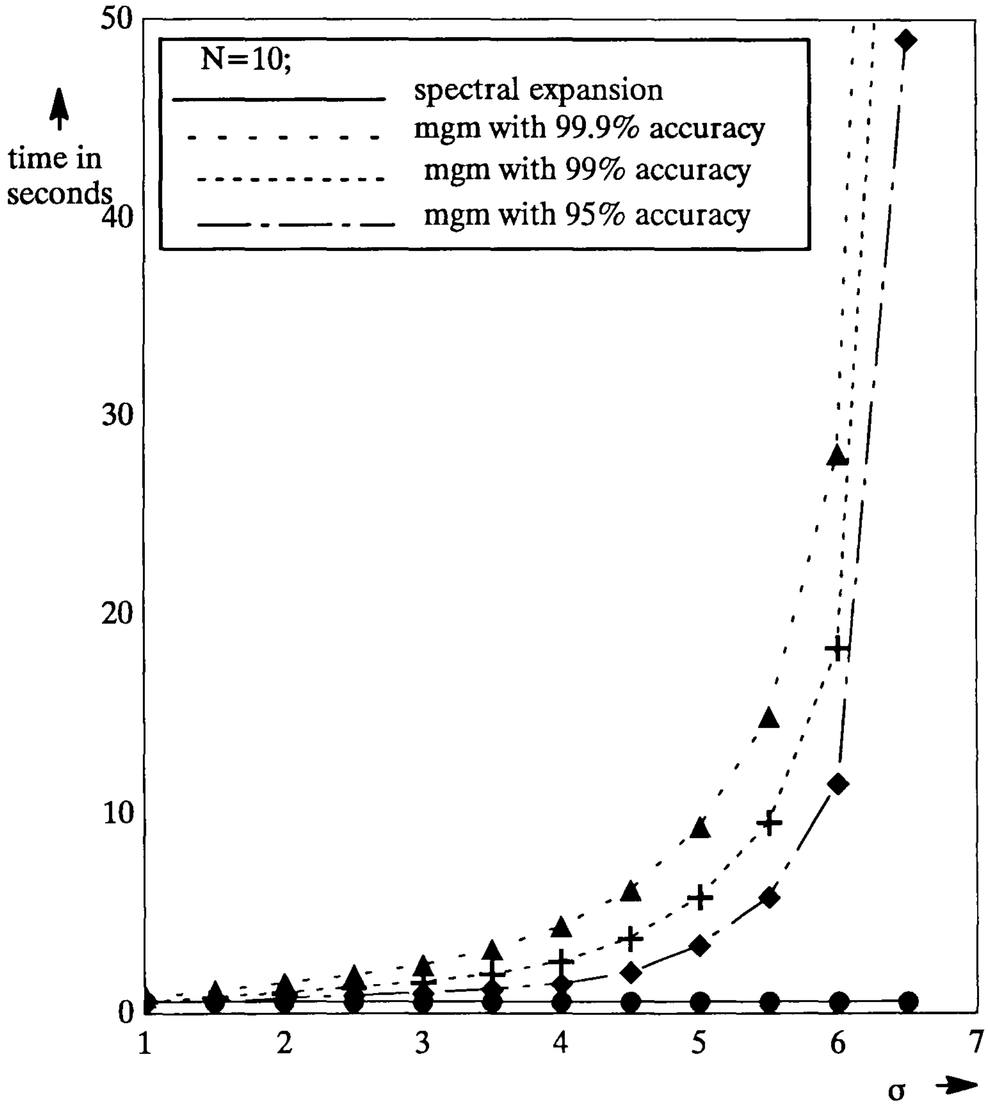

Figure 5.1: Mean queue length computation times for different arrival rates. Service capacity = 6.66667.

The most notable feature of Figure 5.1 is the steep deterioration of the matrix-geometric run time when σ increases. For a heavily loaded system, that run time may be many orders of magnitude larger than the spectral expansion one. The extra time is almost entirely taken up by the procedure that calculates the matrix R. The corresponding numbers of iterations are plotted in Figure 5.2. It seems clear that, even when the accuracy requirements are modest, the computational demands of the iterative procedure have a vertical asymptote at the saturation point. Even here, in all these experiments there is no simple way of knowing before, the number of iterations needed to compute R in order to get performance measures of a pre-specified accuracy.

It can be argued that a different procedure for computing R may yield better results. A number of such procedures have been compared in [25], and improvements have indeed been observed. However, the results reported in [25] indicate that all iterative approaches suffer from the drawbacks exhibited in Table 5.2 and Figure 5.1, namely that their accuracy and efficiency decrease when the load increases. Perhaps the fastest existing method for computing R is not by iterations, but via its eigenvalues and eigenvectors (this was done in [12] for a model with real eigenvalues and a non-singular, diagonal matrix C). If that method is adopted, then one might as well use spectral expansion directly, and avoid the matrix R altogether.

In our experience, using different starting points for the iterations (rather than $R_0 = 0$), does not bring significant benefits and may be dangerous. The procedure could diverge or it could, conceivably, converge to a non-minimal solution of (5.2).

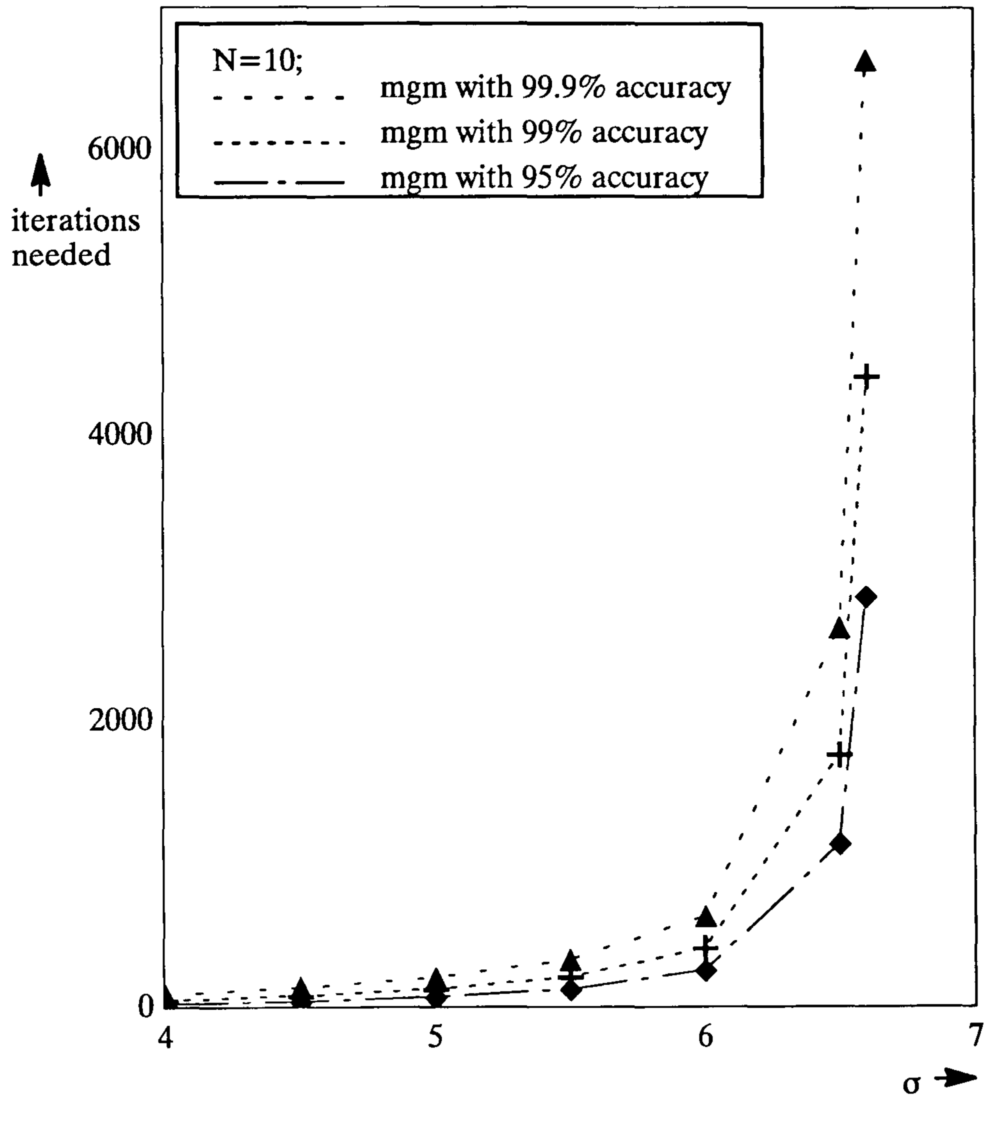

Figure 5.2: Number of iterations needed for computing R, for different arrival rates. Service capacity = 6.66667.

The next set of experiments concerns the behaviour of the two algorithms when the size of the problem, chiefly determined by the number of processors, increases. It is perhaps worth restating the effect of the parameter N.

- Spectral expansion: The characteristic polynomial is of degree at most $2N+2$; finding the eigenvalues and left eigenvectors is a task of complexity $O(N^3)$. The coefficients of the $N+1$ term expansion are determined by solving a set of $N+1$ simultaneous linear equations, which is also a task of complexity $O(N^3)$.

- Matrix-geometric solution: The matrix R is of dimensions $(N+1) \times (N+1)$. Each iteration in the iterative procedure is of complexity $O(N^3)$. Still, the dependency of the number of iterations needed on N, for a desired accuracy, is not known. An unknown vector with $N+1$ elements is determined by solving a set of $N+1$ simultaneous linear equations (complexity $O(N^3)$).

The results of the experiments are displayed in Figures 5.3, 5.4 and 5.5. In all cases, constant load is maintained by increasing the arrival rate in proportion to N. When timing the matrix-geometric runs, the number of iterations for matrix R is chosen so as to achieve 99% accuracy in $E(J)$, relative to the spectral expansion value. Figure 5.3 shows the run times of the two algorithms when the system is quite lightly loaded, ($\sigma = 0.3N$), whereas Figure 5.4 deals with a more heavily loaded queue ($\sigma = 0.6N$). The matrix-geometric run times are greater than the spectral expansion ones by a factor of about 2 in the first case, and by a factor of more than 20 in the second (as before, the spectral expansion times do not depend on the load). This is consistent with the previous observations on the way the iterative procedure is influenced by the offered load. However, if the number of iterations is considered as a function of N, with the load fixed (see Figure 5.5), then the increase is seen to be linear. If that is a general phenomenon, then the overall computational complexity of the iterative procedure for computing R is $O(N^4)$.

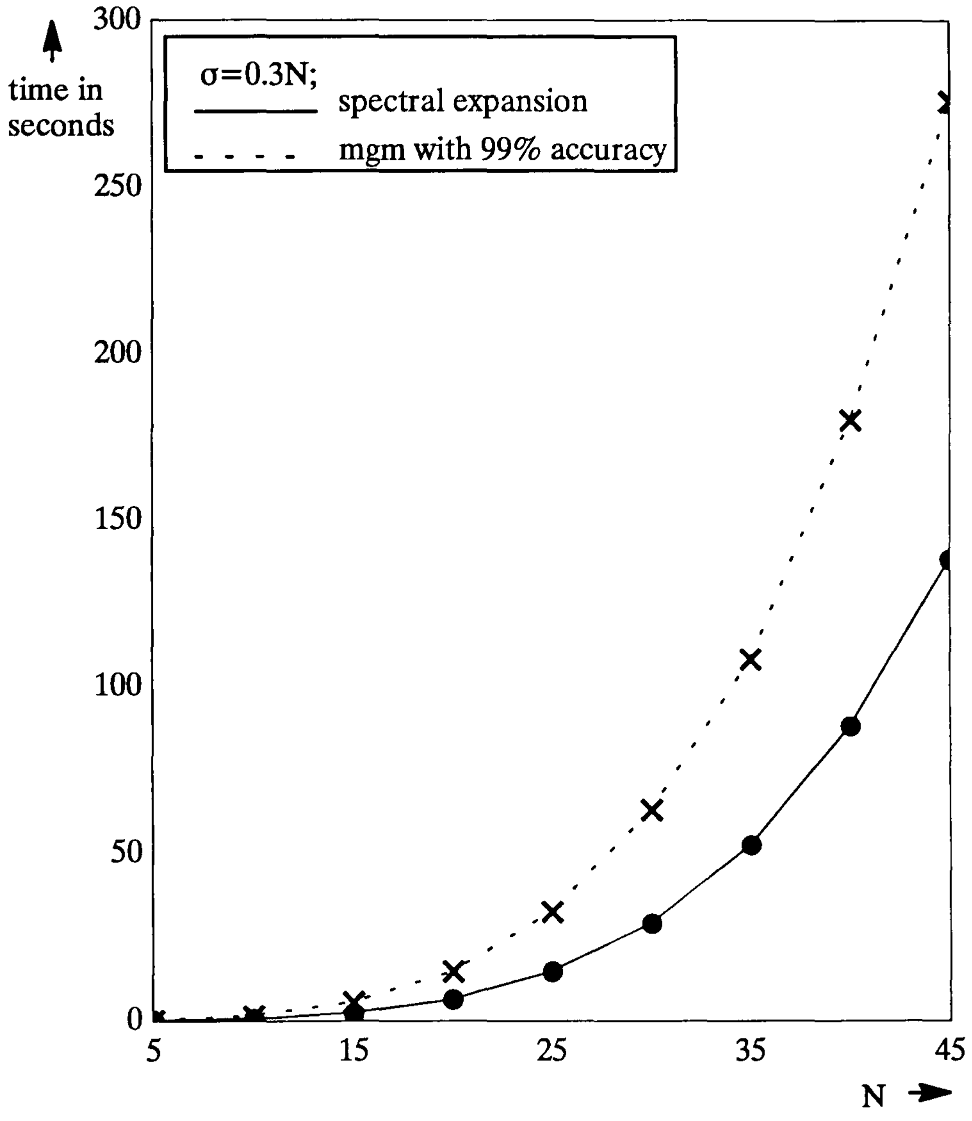

Figure 5.3: Mean queue length computation times for different N. Service capacity $= 0.66667N$.

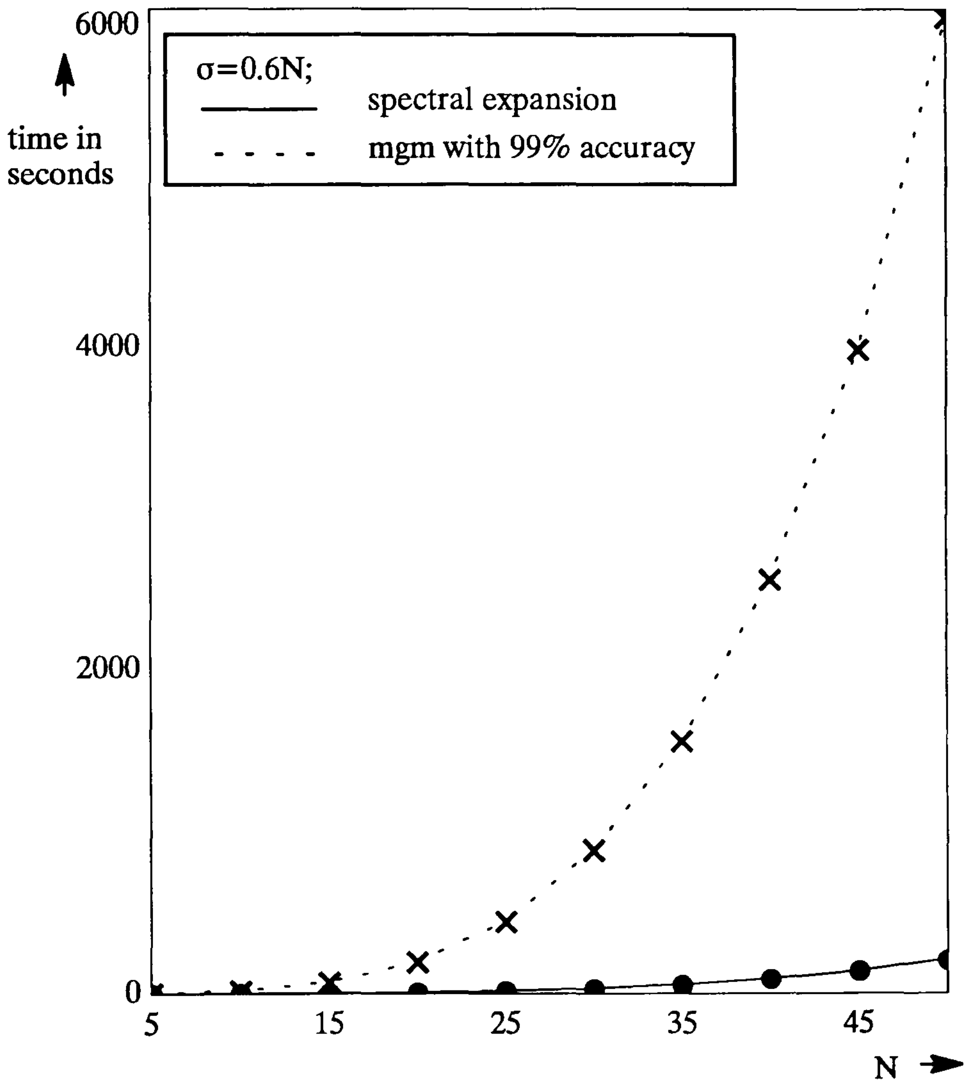

Figure 5.4: Mean queue length computation times for different N. Service capacity $= 0.66667N$.

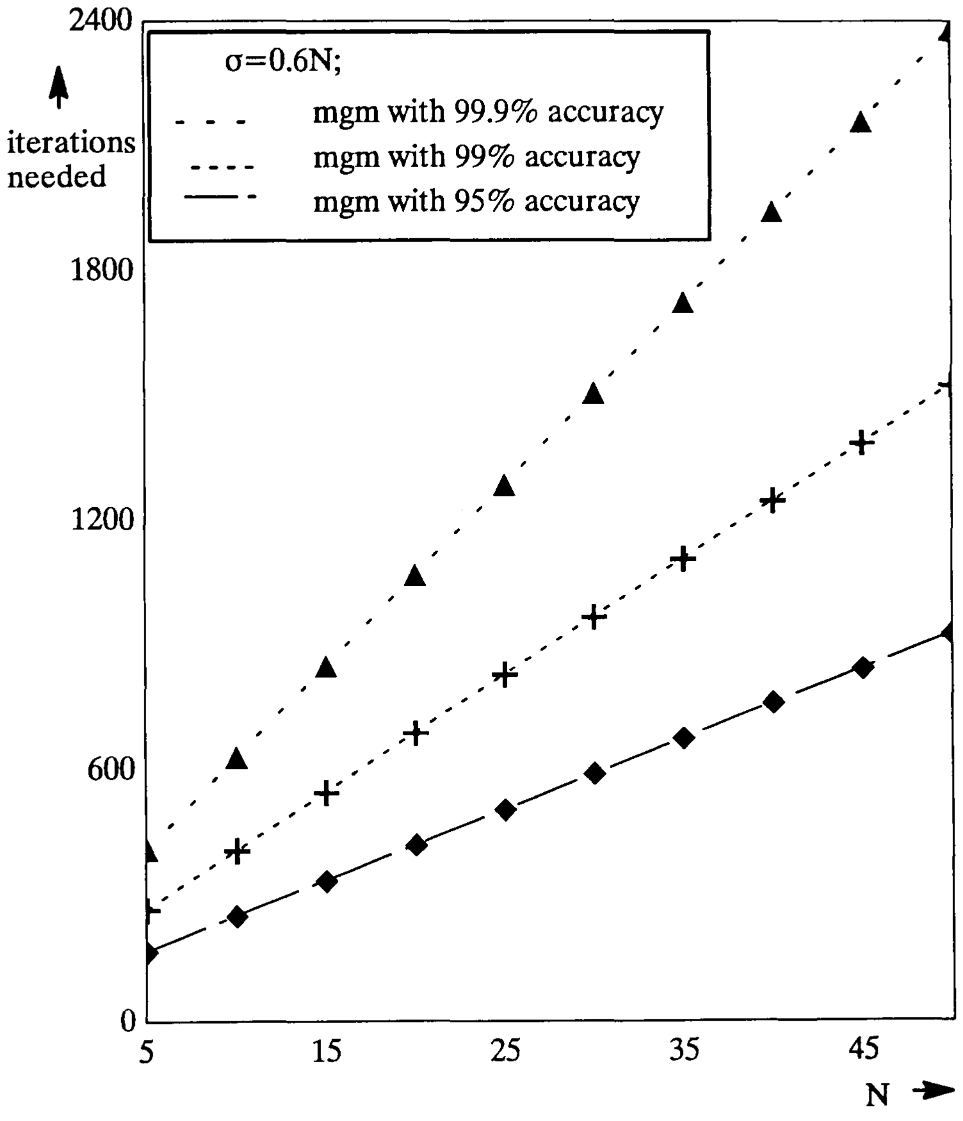

Figure 5.5: Number of iterations needed for computing R, for different N. Service capacity = $0.66667N$.

The second step in both algorithms (solving a set of linear equations), is where emerging numerical problems begin to manifest themselves. The observable symptom of such problems is usually an ill-conditioned matrix. It should be emphasized that neither the spectral expansion nor the matrix-geometric algorithm is immune to ill-conditioned matrices. We have observed them generally when the number of processors is large and the system is lightly loaded. For example, taking $N = 50$, $\sigma < 15$ (other parameters as in the figures), leads to ill-conditioned matrices in both algorithms. This may be due to the limitation of the NAG routine F04ATF. The method used in this routine is, LU factorization with partial pivoting and iterative refinement. Perhaps, there may be more robust methods available, and it is worth trying with them, though we have not done so.

In many cases, the problem of ill-conditioning can be alleviated by appropriate scaling (e.g. multiplying certain vectors by suitable constants). We have tried a few scaling techniques in connection with the spectral expansion algorithm, and have had some success. However, it is a reasonable guess that, within the constraint of double precision arithmetic, neither method can handle accurately models with more than about 100 processors. Of course, that range can be extended considerably by using extended precision, but then the choice of programming language and/or compiler would be restricted.

A question of relevance to the spectral expansion solution concerns its application when two or more eigenvalues coincide, or are very close to each other. We have attempted to construct some examples of this type. Consider a moderately large system with $N = 30$. By examining the 31 eigenvalues in the unit disk for different values of the arrival rate it is noticed that when σ changes from 14.6128383 to 14.6128384, two real eigenvalues, 0.649104533666 and 0.649107441313, change to a pair of complex conjugate ones:

$0.649105990110 - 0.000001630665i$ and $0.649105990110 + 0.000001630665i$.

Hence, there is a value of σ in that interval which would produce a double eigenvalue. The nearest we were able to get to it is the interval:
$\sigma \in (14.61283834428, 14.61283834429)$,
where the two real eigenvalues, 0.649105971264 and 0.649106006036, change to a pair of complex conjugate ones:
$0.649105988650 - 0.000000016211i$ and $0.649105988650 + 0.000000016211i$.
Even in this last case, where the distance between two eigenvalues is on the order of 10^{-7}, the algorithm works well: the equations are not ill-conditioned and a correct value for $E(J)$ is obtained.

Having repeated the above exercise for different other models [10], other than the multiprocessor problem, (e.g. giving rise to a double eigenvalue at 0), with similar results, we feel justified in saying that multiple eigenvalues present an interesting theoretical problem, but not a practical one.

5.4 Conclusions

The Markov models considered in the previous chapters can also be solved by the presently most popular method, the matrix-geometric method. That solution for the model X is presented in this chapter. These two solutions, viz. spectral expansion and the matrix-geometric method, are compared through the example of a multiprocessor system, for compuational efficiency, accuracy and numerical stability.

When comparing the spectral expansion and the matrix-geometric solutions, a strong case can be made for the former. Spectral expansion offers considerable advantages in efficiency, without undue penalties in terms of numerical stability. The speed gains are especially pronounced, and indeed appear to be unbounded, when the system approaches saturation. However, further comparisons, perhaps with different methods for computing R, or with other methods such as Seelen's [55], would be desirable. It would also be interesting to compare the different methods on problems where the jumps of J are greater than 1.

The numerical stability in all our experiments has been limited partially because of the robustness and accuracy limitations of the NAG routines we have used. We believe, that numerical stability can be improved further by resorting to more accurate and more robust algorithms and procedures, for eigenvalue-eigenvector computation and for solution of linear simultaneous equations, available in the literature. However, that experimentation is beyond the scope of this thesis.

5.5 Contributions

The contribution of this chapter is,

- The matrix-geometric method for the QBD process X is presented. A detailed comparison study is made between the spectral expansion method and the matrix-geometric method. This is done through the example of a homogeneous multiprocessor system with general failures and repairs, serving a Poisson stream of jobs. Meticulous experiments are performed for comparing computational efficiency, accuracy and numerical stability of the methods. A strong case is established for the spectral expansion method, over the matrix-geometric method.

This contribution is to appear in our paper [47]. Part of these results have been reported in our paper [8].

Chapter 6

Heterogeneous Multiprocessor Systems with Breakdowns: Performance and Optimal Repair Strategies

6.1 Introduction

Several non-trivial example systems have been modelled and solved in the previous chapters using spectral expansion. It is now time to turn to even more complex systems. These systems may need, in addition to spectral expansion, several other systems engineering techniques for an in-depth analysis and solution. Such are the systems considered in this chapter, and in the later ones. This chapter deals with a heterogeneous multiprocessor system serving an incoming job stream. The multiprocessor has K non-identical processors that are prone to breakdowns and repairs.

The general topic of modelling systems which are subject to interruptions of service has received considerable attention in the literature. Some of the work has concentrated on single-processor models [1, 19, 57, 62, 67], and some on multiprocessor ones where all the processors are statistically identical [5, 42, 44, 50]. However, very little progress has been made in modelling heterogeneous multiprocessor systems with breakdowns. Although Markov-modulated queueing systems [53, 69] provide a rather general framework in which this problem may be placed, there are no existing solutions. Some of the difficulty of the analysis stems from the fact that, at any moment in time, the total rate at which service is given depends both on the state of the processors and, to a limited extent, on the number of jobs present.

This chapter has a two-fold contribution. First, we study the problem and model the system by the Markov model X. This is done in the next section. In section 6.3, we solve that system using the spectral expansion method, for steady state performance and dependability measures.

The second contribution concerns optimization. If the number of repairmen is less than K, thus restricting the number of processors that can be repaired in parallel, then the repair scheduling strategy (i.e. the order in which broken processors are selected for repair), has an important effect on performance. In section 6.4, we study that effect at some length, concentrating on the case of a single repairman. An interesting problem which, to our knowledge, has not been addressed before, is to find the repair strategy that maximizes the average processing capacity of the system. A related problem was considered and solved by Kameda [32] in the context of a finite-source queue with different customers. This is regarding the maximization of a weighted sum of the utilisation factors of the CPU, for different classes of jobs, in the finite-source queue, under a restricted condition. That problem is equivalent to maximizing a weighted sum of the utilizations of the repairman, for different processors, in our model, when the breakdown rates of all processors are equal. We use the results of Kameda, to find the optimal capacity strategy for the same special case.

The general optimal capacity problem still lacks an exact solution. We propose several heuristics and examine their performance empirically. This is done in section 6.5

6.2 The model

Jobs arrive into the system in a Poisson stream at rate σ, and join an unbounded queue. There are K non-identical parallel processors numbered $1, 2, \ldots, K$. The service times of jobs executed on processor k ($k = 1, 2, \ldots, K$) are distributed exponentially with mean $1/\mu_k$. However, processor k executes jobs only during its operative periods, which are distributed exponentially with mean $1/\xi_k$. At the end of an operative period, processor k breaks down and requires an exponentially distributed repair time with mean $1/\eta_k$. The number of repairs that may proceed in parallel could be restricted: if so, this is expressed by saying that there are R repairmen ($R \leq K$), each of whom can work on at most one repair at a time. Thus, an inoperative period of a processor may include waiting for a repairman. No operative processor can be idle if there are jobs awaiting service, and no repairman can be idle if there are broken down processors. All inter-arrival, service, operative and repair time random variables are independent of each other.

If there are more operative processors than jobs in the system, then the busy processors are selected according to some *service priority* ordering. That ordering can be arbitrary, but it is clearly best to execute the jobs on the processors whose service rates are highest. Such a result, but in the absence of breakdowns, has been proved in [39]. This is therefore assumed to be the case, unless specified otherwise. It is further assumed that jobs can migrate instantaneously from processor of lower service priority to processor of higher service priority, in the middle of a service, if the latter is available. Services that are interrupted by breakdowns are eventually resumed (perhaps on a different processor and hence at a different rate), from the point of interruption (or, it can be *repeat with re-sampling*, both these service interruption policies are equivalent). Similarly, if $R < K$ and the repair strategy allows preemptions of repairs, then those are eventually resumed from the point of interruption and there are no switching delays.

The system is illustrated in Figure 6.1.

The system state at time t can be described by a pair of integers, $I(t)$ and $J(t)$, specifying the processor configuration (can also be termed, operative state of the multiprocessor) and the number of jobs present, respectively. The precise meaning of "processor configuration", and hence the range of values of $I(t)$, depend on the assumptions. For example, if the processors are identical, as in the case of the example in Chapter 3, then it is sufficient to specify how many of them are operative: $I(t) = 0, 1, \ldots, K$. In the heterogeneous case, if $R = K$, or if the repair strategy is pre-emptive priority or processor sharing, the processor configuration should specify, for each processor, whether it is operative or broken down. This can be done by using a range of values $I(t) = 0, 1, \ldots, 2^K - 1$ (e.g., each bit in the K-bit binary expansion of $I(t)$ can indicate the state of one processor). If, on the other hand, $R < K$ and the repair strategy is non-preemptive, then it is necessary to specify both the processors that are broken down and the ones among them that are being repaired. That would require an even wider range of values for $I(t)$.

In general, let there be $N + 1$ processor configurations, represented by the values $I(t) = 0, 1, \ldots, N$. These $N + 1$ configurations are the operative states of the model. The model assumptions ensure that $I(t)$, $t \geq 0$, is an irreducible Markov process. $J(t)$ is the total number of jobs in the system at time t, including the ones in service. Then, $X = \{[I(t), J(t)]; t \geq 0\}$ is an irreducible Markov process that models the system. Its state space is, $\{0, 1, \ldots, N\} \times \{0, 1, \ldots\}$.

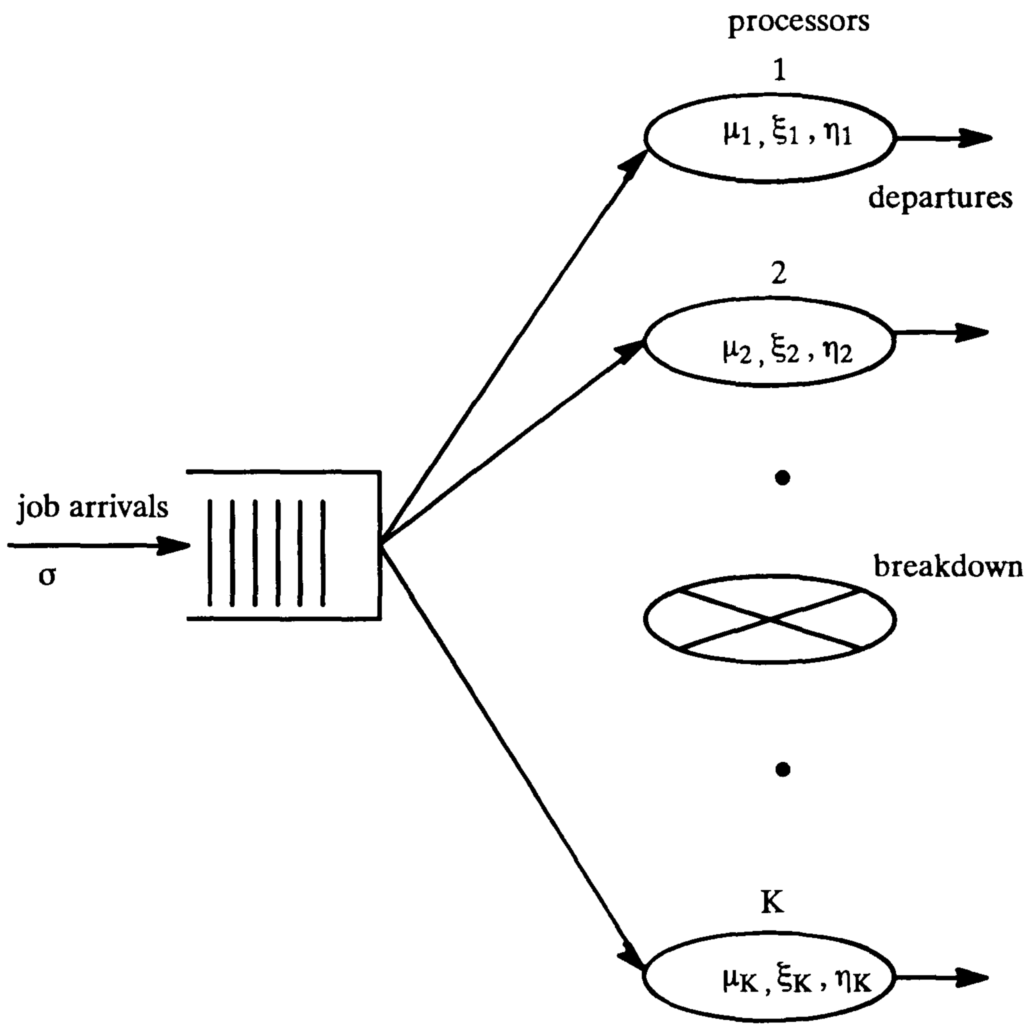

Figure 6.1: Heterogeneous multiprocessor system with a common unbounded queue.

Let A_j be the matrix of instantaneous transition rates from state (i,j) to state (r,j) for that process $(i,r = 0,1,\ldots,N\ ;\ i \neq r)$, with zeros on the main diagonal. These are the purely lateral transitions of the model X. The elements of A_j depend on the parameters ξ_k and η_k $(i = 1,2,\ldots,K)$ and, if $R < K$, on the repair strategy also. For example, consider a system with 3 processors and 1 repairman, under a preemptive priority repair strategy with priority ordering $(1,2,3)$ (i.e., processor 1 has top preemptive priority for repair, while processor 3 has bottom priority). There are now 8 processor configurations, $I(t) = 0,1,\ldots,7$, representing the operative states $(0,0,0),(0,0,1),\ldots,(1,1,1)$ respectively (k^{th} bit, from left, is 0 when processor k is broken, 1 when operative). In this case, the matrix A and A_j are given by

$$\begin{array}{c} (0,0,0) \\ (0,0,1) \\ (0,1,0) \\ (0,1,1) \\ (1,0,0) \\ (1,0,1) \\ (1,1,0) \\ (1,1,1) \end{array} \quad A = A_j = \begin{bmatrix} 0 & 0 & 0 & 0 & \eta_1 & 0 & 0 & 0 \\ \xi_3 & 0 & 0 & 0 & 0 & \eta_1 & 0 & 0 \\ \xi_2 & 0 & 0 & 0 & 0 & 0 & \eta_1 & 0 \\ 0 & \xi_2 & \xi_3 & 0 & 0 & 0 & 0 & \eta_1 \\ \xi_1 & 0 & 0 & 0 & 0 & 0 & \eta_2 & 0 \\ 0 & \xi_1 & 0 & 0 & \xi_3 & 0 & 0 & \eta_2 \\ 0 & 0 & \xi_1 & 0 & \xi_2 & 0 & 0 & \eta_3 \\ 0 & 0 & 0 & \xi_1 & 0 & \xi_2 & \xi_3 & 0 \end{bmatrix} \ ; j \geq 0 . \qquad (6.1)$$

Instead, if the repair strategy is one of static non-preemptive priority, it can be shown that the matrix A_j would be of size 13×13, for the 3-processor system. Those 13 processor configurations are, are: $(0,0,0)_1$, $(0,0,0)_2$, $(0,0,0)_3$, $(0,0,1)_1$, $(0,0,1)_2$, $(0,1,0)_1$, $(0,1,0)_3$, $(1,0,0)_2$, $(1,0,0)_3$, $(0,1,1)$, $(1,0,1)$, $(1,1,0)$ and $(1,1,1)$. Here, the suffix indicates the index of the broken processor that is being repaired, if the number of broken processors is more than 1.

FCFS repair strategy also can be modelled. That would need much more number of processor configurations. In the case of $K = 3$, those configurations, 16 in number, are: $(0,0,0)_{1,2,3}$, $(0,0,0)_{1,3,2}$, $(0,0,0)_{2,1,3}$, $(0,0,0)_{2,3,1}$, $(0,0,0)_{3,1,2}$, $(0,0,0)_{3,2,1}$, $(0,0,1)_{1,2}$, $(0,0,1)_{2,1}$, $(0,1,0)_{1,3}$, $(0,1,0)_{3,1}$, $(1,0,0)_{2,3}$, $(1,0,0)_{3,2}$, $(0,1,1)$, $(1,0,1)$, $(1,1,0)$ and $(1,1,1)$. Here, the suffix indicates the order in which the broken processors are to be repaired, if their number is more than 1. For several other kinds of repair strategies, it is possible to work out the A_j matrix quite easily, if not for all possible repair strategies.

It can be seen that the transitions in the processor configuration are not accompanied by changes in the number of jobs, unless the latter is an independent coincidence. Hence, for all repair strategies and number of repairmen, it can be assumed, $A_j = A$; $j \geq 0$. Let also D^A be, as defined in Chapter 2, the diagonal matrix whose i^{th} diagonal element is the i^{th} row sum of A. The matrix $A - D^A$ is the *generator matrix* of the process $I(t)$.

Denote by q_i the steady-state probability that the processor configuration is i. The vector $\mathbf{v} = (q_0, q_1, \ldots, q_N)$ is determined by solving the linear equations,

$$\mathbf{v}(A - D^A) = \mathbf{0} \ ; \ \mathbf{v}\mathbf{e} = 1 \ , \tag{6.2}$$

where $\mathbf{e}$ is the column vector with $N+1$ elements, all of which are equal to 1. That solution is not too expensive, even for large values of N, because the matrix A is very sparse. Having obtained $\mathbf{v}$, one can find the steady-state probability, α_k, that processor k is operative. It is given by

$$\alpha_k = \sum_{i \in S_k} q_i \ ; \ k = 1, \ldots, K \ , \tag{6.3}$$

where S_k is the set of processor configurations in which processor k is operative. The linear combination

$$\gamma = \sum_{k=1}^{K} \alpha_k \mu_k \tag{6.4}$$

will be referred to as the *processing capacity* or service capacity of the system. This is the maximum average rate at which jobs can be executed. It is then intuitively clear that the system is stable if and only if $\sigma < \gamma$. However, the rigorous proof of this statement is beyond the scope of this thesis.

In order to compute performance measures involving jobs, it is necessary to study the two-dimensional Markov process, X. Assuming that the stability condition holds, our next task is to compute the steady-state probabilities, $p_{i,j}$, that the processor configuration is i and the number of jobs in the system is j. Let, as in Chapter 2, the vector $\mathbf{v}_j = (p_{0,j}, p_{1,j}, \ldots, p_{N,j})$.

6.3 Spectral expansion solution

As we have seen in the previous section, the process X models the given system exactly. X is the Quasi-Birth-and-Death process described in Chapter 2, as we examine from its parameters. The one-step upward transition rate matrices, B and B_j, of size $(N+1) \times (N+1)$, are given by,

$$B_j = B = diag[\sigma, \sigma, \ldots, \sigma] \; ; \; j \geq 0 \, . \qquad (6.5)$$

Let the one-step downward transition rate, $C_j(i,r)$, from state (i,j) to state $(r, j-1)$, $j \geq 1$, is equal 0 if $i \neq r$, and $\gamma_{i,j}$ if $i = r$. This is because when a job departs there is no reason why the processor configuration should change, unless the latter is an independent coincidence. $C_j(i,i)$ or $\gamma_{i,j}$ is the *average processing rate* of the system when $I(t) = i$ and $J(t) = j$. $C_j(i,i)$ or $\gamma_{i,j}$, when $j \geq K$, is the *processing capacity of the configuration* i, and does not depend on j. Hence, let $\gamma_{i,j} = \gamma_i$ for $j \geq K$. It is possible to work out $C_j(i,i)$ or $\gamma_{i,j}$ for $j < K$. This is done below, for the 8 processor configurations of the 3-processor example with pre-emptive priority repair. Assuming that, $\mu_1 \geq \mu_2 \geq \mu_3$ and service priority order is (1, 2, 3), the values of $\gamma_{i,j}$ are,

configuration	$\gamma_{i,1}$	$\gamma_{i,2}$	$\gamma_{i,j}(j \geq 3) = \gamma_i$
$(0,0,0)$	0	0	0
$(0,0,1)$	μ_3	μ_3	μ_3
$(0,1,0)$	μ_2	μ_2	μ_2
$(0,1,1)$	μ_2	$\mu_2 + \mu_3$	$\mu_2 + \mu_3$
$(1,0,0)$	μ_1	μ_1	μ_1
$(1,0,1)$	μ_1	$\mu_1 + \mu_3$	$\mu_1 + \mu_3$
$(1,1,0)$	μ_1	$\mu_1 + \mu_2$	$\mu_1 + \mu_2$
$(1,1,1)$	μ_1	$\mu_1 + \mu_2$	$\mu_1 + \mu_2 + \mu_3$

Hence, the matrices C_j and C are given by,

$$\begin{aligned} C_j &= C = diag[\gamma_0, \gamma_1, \ldots, \gamma_N] \; ; \; j \geq K, \\ C_j &= diag[\gamma_{0,j}, \gamma_{1,j}, \ldots, \gamma_{N,j}] \; ; \; 1 \leq j \leq K, \\ C_0 &= 0 \, . \end{aligned} \tag{6.6}$$

The threshold, M, of this system is given by, $M = K$. Note that, this threshold should be the same for any repair strategy, since the the average processing rate, $\gamma_{i,j}$ in a given configuration i with a given number of jobs j, does not depend on j if $j \geq K$.

D^A, D^A_j, D^B, D^B_j, D^C and D^C_j are the diagonal matrices, defined as in Chapter 2, with respect to A, A_j, B, B_j, C and C_j respectively. And, $Q_0 = B$, $Q_1 = A - D^A - D^B - D^C$, and $Q_2 = C$.

Then the valid spectral expansion for $\mathbf{v}_j$ is,

$$\mathbf{v}_j = \sum_{i=0}^{N} a_i \psi_i \lambda_i^j \; ; \; j \geq K - 1 \, . \tag{6.7}$$

where, (λ_i, ψ_i), $i = 0, 1, \ldots, N$, are the eigenvalue-eigenvectors of $Q(\lambda)$ that (eigenvalues) are inside the unit circle(when the ergodicity condition is satified).

The system can be solved for the required $\mathbf{v}_j$'s and steady state performance measures, using the procedure in Chapter 2.

Figure 6.2 shows the plot of average number of jobs in the system, $E(J)$, as a function of the job arrival rate σ. This is done for a 3-processor system with a single repairman, for several different repair strategies.

It may be noted, the system can be solved even if the job arrival rates depend on the processor configuration i. In such a case, the matrices B and B_j's need to be modified accordingly. It is also possible to model systems in which a job is lost when its service is interrupted by the server failure. That requires a modification of the matrices C and C_j's accordingly.

It is possible to incorporate job arrivals in *batches* of bounded sizes. That requires modelling, by the process Y, permitting bounded multi-step jumps in the random variable J of Y. However, numerical examples of this type are not dealt in this thesis.

6.4 Repair strategy optimization

When the number of repairmen is less than the number of processors, the order in which processors are selected for repair has an influence on the performance of the system. Different repair strategies can lead to different values of the processing capacity, γ, and hence to arbitrarily large differences in performance measures like the average queue size or the average response time. This situation is illustrated in Figure 6.2.

In this figure, the average number of jobs, $E(J)$, in a 3-processor system with one repairman, is plotted against the job arrival rate, σ. Five different repair strategies are compared, three of which are of the preemptive-resume type, allowing repairs to be interrupted at an arbitrary point without any overheads. One of the remain-

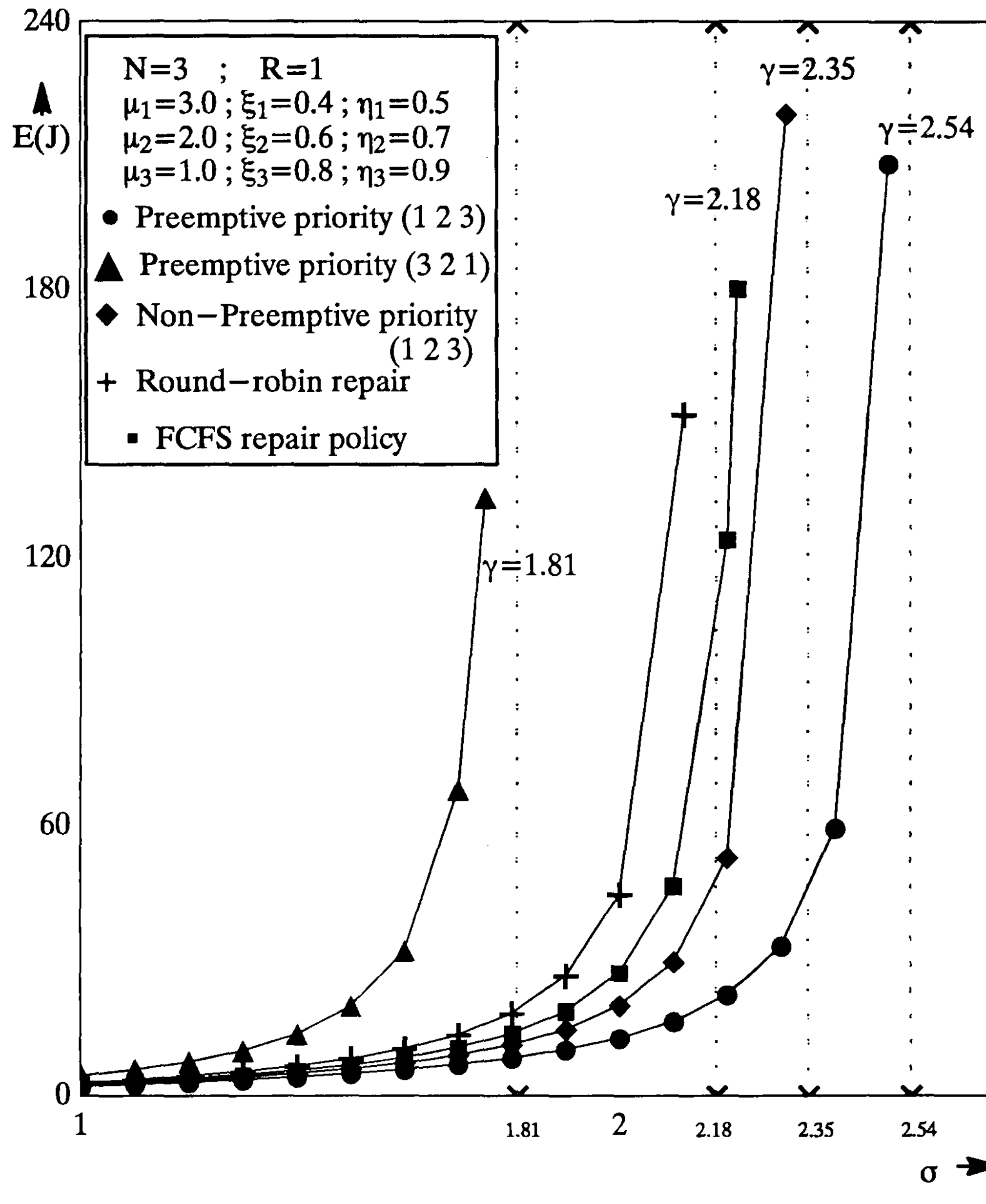

Figure 6.2: Importance of repair strategy: A 3-processor system example.

ing two is of non-preemptive type with priorities, and the other is of FCFS repair strategy. In all cases, the values of $E(J)$ are determined by applying the spectral expansion procedure described in the previous section.

The parameters in this example may have not been chosen to be realistic; nevertheless, the behaviour that is displayed in the figure is quite typical. The important feature of all such performance curves is that they have a vertical asymptote at the point where the arrival rate is equal to the processing capacity. Any difference in γ, no matter how small, caused by a change in the repair strategy, implies different vertical asymptotes. Hence, given a sufficiently heavy traffic, there will be unbounded differences in performance. In particular, on the interval between the two processing capacities pertaining to two different repair strategies, the system is non-ergodic and the queue becomes saturated under one strategy and stable under the other.

Clearly, it can be very important to use a repair strategy that maximizes the processing capacity. We shall consider the problem of finding such strategies, in the case where K heterogeneous processors are attended to by a single repairman ($R = 1$). This is obviously the case with the greatest practical relevance, since increasing the number of repairmen can only reduce the effect of the repair strategy on the processing capacity.

In the next section, we take the case of a two-processor system, and later go on to the general K-processor system.

6.4.1 Two-processor system

A system, with two different processors, is considered. The question is to find the repair strategy that maximizes γ. Three different repair strategies are considered. They are, (i) preemptive priority, (ii) non-preemptive priority or HOL(head on line), and (iii) generalised round robin repair strategy.

Preemptive priority

The processors are numbered as 1 and 2. Let the processing capacities under the two preemptive priority strategies $(1,2)$ and $(2,1)$ be $\gamma^{(1,2)}$ and $\gamma^{(2,1)}$, respectively. By solving equations (6.2) explicitly (there are now 4 processor configurations: $(0,0)$, $(0,1)$, $(1,0)$ and $(1,1)$), and then using equations (6.4), we find

$$\gamma^{(1,2)} = \frac{\eta_1}{\xi_1 + \eta_1}\left[\mu_1 + \frac{\mu_2\eta_2(s-\eta_2)}{\xi_2 s + \eta_1\eta_2}\right], \tag{6.8}$$

where $s = \xi_1 + \xi_2 + \eta_1 + \eta_2$. The expression for $\gamma^{(2,1)}$ is similar, with the indices 1 and 2 interchanged. A little further manipulation yields

$$\gamma^{(1,2)} - \gamma^{(2,1)} = \frac{\xi_1\xi_2 s}{(\xi_1+\eta_1)(\xi_2+\eta_2)}\left[\frac{\mu_1\eta_1}{\xi_1 s + \eta_1\eta_2} - \frac{\mu_2\eta_2}{\xi_2 s + \eta_1\eta_2}\right]. \tag{6.9}$$

Thus, when $K = 2$, the optimal priority allocation, in the case of preemptive priority, depends on the quantity

$$d_{1,2} = \frac{\mu_1\eta_1}{\xi_1 s + \eta_1\eta_2} - \frac{\mu_2\eta_2}{\xi_2 s + \eta_1\eta_2}. \tag{6.10}$$

Processor 1 should have higher repair priority than processor 2 if $d_{1,2} > 0$, and lower priority if $d_{1,2} < 0$.

HOL

In this repair strategy, when a broken processor is allocated to the repairman, that repair continues till it is completed, without preemption. Here, there are 5 processor configurations. These can be represented as, $(0,0)_1$, $(0,0)_2$, $(0,1)$, $(1,0)$, $(1,1)$. $(0,0)_1$ means both the processors are broken and processor 1 is being repaired. In $(0,0)_2$, processor 2 is being repaired, instead.

Here again, it is possible to find the q_i's and α_k's by solving equations (6.2), and the processing capacity $\gamma^{(HOL)}$ by equation (6.4). When that is done, we have,

$$\gamma^{(HOL)} = \frac{\mu_2\eta_2(\eta_1+\xi_1) + \mu_1\eta_1(\eta_2+\xi_2)}{(\eta_1+\xi_1)(\eta_2+\xi_2) + \xi_1\xi_2}. \tag{6.11}$$

Now, it is obviously interesting to see how $\gamma^{(HOL)}$ compares with the best of $\gamma^{(1,2)}$ and $\gamma^{(2,1)}$. For this, let us assume $\gamma^{(1,2)} > \gamma^{(2,1)}$, by assuming $d_{1,2} > 0$. Then, from equations (6.8, 6.11), we have,

$$\gamma^{(1,2)} - \gamma^{(HOL)} = \frac{\xi_1\xi_2[\mu_1\eta_1(\xi_2 s + \eta_1\eta_2) - \mu_2\eta_2(\xi_1 s + \eta_1\eta_2)]}{(\eta_1 + \xi_1)[(\eta_1 + \xi_1)(\eta_2 + \xi_2) + \xi_1\xi_2](\eta_1\eta_2 + \xi_2 s)} \tag{6.12}$$

Since $d_{1,2} > 0$, from equation (6.10), it implies that $[\mu_1\eta_1(\eta_1\eta_2+\xi_2 s)-\mu_2\eta_2(\eta_1\eta_2+\xi_1 s)] > 0$. Hence, $\gamma^{(1,2)} - \gamma^{(HOL)} > 0$.

That proves, the best preemptive repair policy is better than the non-preemptive repair policy. Note that, in the case of $K = 2$, the FCFS repair strategy is the same as the HOL repair strategy.

Generalised round robin repair strategy

We rather consider the *generalised processor sharing* repair strategy as an approximation to the generalised round robin repair strategy. In this repair strategy, when both the processors are broken, the repairman works on both of them simultaneously. This is done by allocating a fraction, x $(0 \le x \le 1)$, of repairman's time, for repairing processor 1 and the rest, i.e. fraction $(1 - x)$, is allocated to the repair of processor 2. Here too, there are 4 processor configurations, $(0,0)$, $(0,1)$, $(1,0)$, $(1,1)$. But the transition rates are different from those of the preemptive priority repair case. In this case too, by solving equations (6.2, 6.4), the processing capacity, $\gamma^{(grr)}$ is worked out. That is given by,

$$\gamma^{(grr)} = \frac{x\eta_1[\mu_1(\xi_2 s + \eta_1\eta_2) + \mu_2\eta_2(s - \eta_2)] + (1 - x)\eta_2[\mu_2(\xi_1 s + \eta_1\eta_2) + \mu_1\eta_1(s - \eta_1)]}{x\eta_1(\xi_2 s + \eta_1\eta_2 + \eta_2\xi_1) + (1 - x)\eta_2(\xi_1 s + \eta_1\eta_2 + \eta_1\xi_2) + \xi_1\xi_2 s}. \tag{6.13}$$

Now let us compare $\gamma^{(grr)}$ with $\gamma^{(1,2)}$ assuming, $d_{1,2} > 0$.

Differentiating $\gamma^{(grr)}$ with respect to x, we get,

$$\frac{\partial}{\partial x}\{\gamma^{(grr)}\} = \frac{\xi_1\xi_2 s[\mu_1\eta_1(\xi_2 s + \eta_1\eta_2) - \mu_2\eta_2(\xi_1 s + \eta_1\eta_2)]}{[x\eta_1(\xi_2 s + \eta_1\eta_2 + \eta_2\xi_1) + (1-x)\eta_2(\xi_1 s + \eta_1\eta_2 + \eta_1\xi_2) + \xi_1\xi_2 s]^2} \tag{6.14}$$

As can be seen, $\frac{\partial}{\partial x}\{\gamma^{(grr)}\}$ is always positive, since $d_{1,2} > 0$. Hence, $\gamma^{(grr)}$ is an increasing function of x, its minimum is when $x = 0$, and maximum at $x = 1$. However, when $x = 1$, then $\gamma^{(grr)} = \gamma^{(1,2)}$ and the round robin service becomes the same as preemptive priority $(1, 2)$.

Hence, it may be said, the best repair strategy, for $K = 2$, is one of the preemptive priority ones.

6.4.2 K-processor system

Let us now come back to the K-processor system. For a given strategy, let u_k be the fraction of time that processor k spends being repaired ($k = 1, 2, \ldots, K$); alternatively, this is the repairman utilisation factor due to processor k. Let also T_k be the average period between two consecutive instants when processor k becomes operative (i.e. the average *operative-inoperative* cycle for processor k). Then we can write

$$u_k = \frac{1/\eta_k}{T_k} \; ; \; k = 1, 2, \ldots, K \; . \tag{6.15}$$

Similarly, the probability that processor k is operative can be written as

$$\alpha_k = \frac{1/\xi_k}{T_k} \; ; \; k = 1, 2, \ldots, K \; . \tag{6.16}$$

Hence, we have the relation $\alpha_k = (\eta_k/\xi_k)u_k$. This allows us to rewrite the expression for the processing capacity (6.4) in the form

$$\gamma = \sum_{k=1}^{K} c_k u_k \; , \tag{6.17}$$

where $c_k = \mu_k\eta_k/\xi_k$. Thus, the problem of maximizing the processing capacity can be reduced to that of maximizing a linear combination of repairman utilisations. This latter problem has been studied by Kameda [32, 33], in the context of a finite-source queue with different jobs, multiple peripherals and a single CPU. He considered the special case where the job service rates at the peripherals (which correspond to our breakdown rates of processors) are equal: $\xi_1 = \xi_2 = \ldots = \xi_K$. Under that assumption, and using our terminology, Kameda's main result can be summarized as follows (i.e. when the breakdown rates of processors are equal):

(a) When the number of repairmen is 1 ($R = 1$, is already assumed), and the repair rates of different servers can be different, then the strategy that maximizes the right-hand side of (6.17) is one of the $K!$ static preemptive priority strategies.

(b) In the optimal preemptive priority strategy, processor k is given higher priority for repair than processor m if $c_k > c_m$.

Thus, when the breakdown rates are equal, the highest processing capacity is achieved by a static repair strategy which gives processor k preemptive priority over processor m, if $\mu_k \eta_k > \mu_m \eta_m$.

In the general case of unequal breakdown rates, it may be that, in order to achieve global optimality, one has to allow dynamic scheduling decisions which depend in a complex way on the current processor configuration. Nevertheless, we shall confine our search to the set of $K!$ static preemptive priority policies. Even so, it seems that there is no simple rule for determining the optimal allocation of priorities. A straightforward application of the recipe contained in statement (b), to the general case, leads to the following:

Heuristic 1 *Give processor k preemptive priority for repair over processor m, if $\mu_k \eta_k / \xi_k > \mu_m \eta_m / \xi_m$.*

This heuristic seems to be a reasonable guess, even considering the equation (6.17) that is valid for the general case. The thing this heuristic suggests is, to allocate the repairman to the broken processor with the largest value of c_k, that too on a pre-emptive basis, only with a hope that might maximize the r.h.s. of equation (6.17).

Other heuristics can be suggested, based on the results obtained on the two-processor system, discussed earlier. It has been shown there, that the preemptive priority has been the best. That result suggests some heuristic ways of ranking the processors in a general K-processor system. Define, for any pair of processors k and m, the quantity

$$d_{k,m} = \frac{\mu_k \eta_k}{\xi_k s_{k,m} + \eta_k \eta_m} - \frac{\mu_m \eta_m}{\xi_m s_{k,m} + \eta_k \eta_m} , \tag{6.18}$$

where $s_{k,m} = \xi_k + \xi_m + \eta_k + \eta_m$. Also define the indicator

$$\delta_{k,m} = \begin{cases} 1 & if \quad d_{k,m} > 0 \\ 0 & if \quad d_{k,m} \leq 0 \end{cases}$$

Processor k is said to be *preferable* to processor m if $\delta_{k,m} = 1$.

Two heuristics based on this notion of preference are:

Heuristic 2 *Assign to processor k an integer weight, w_k, equal to the number of processors to which it is preferable:*

$$w_k = \sum_{m=1}^{K} \delta_{k,m} \quad ; \quad k = 1, 2, \ldots, K .$$

Give processor k higher priority than processor m if $w_k > w_m$.

Heuristic 3 *Assign to processor k a real weight, w_k, equal to the accumulated differences (20), for all processors to which it is preferable:*

$$w_k = \sum_{m=1}^{K} \delta_{k,m} d_{k,m} \quad ; \quad k = 1, 2, \ldots, K .$$

Give processor k higher priority than processor m if $w_k > w_m$.

Heuristic 2 concentrates on the existence of preference relations, whereas Heuristic 3 attempts to take into account their *extent*. Of course, both produce the optimal allocation in the case $K = 2$.

The relation *preferable* is not transitive. For example, it may happen that processor i is preferable to processor j, processor j is preferable to processor k, and processor k is preferable to processor i. Heuristic 2 could then assign equal weights to those three processors, necessitating some tie-breaking rule. We have simply used the processor indices to break ties. It should be pointed out that these occurrences are very rare (less than 1% of all cases examined).

One can put forward other heuristics which, while being intuitively weaker, do not seem unreasonable. For example:

Heuristic 4 *Give processor k priority over processor m, if* $\mu_k \eta_k > \mu_m \eta_m$.

Heuristic 5 *Give processor k priority over processor m, if* $\mu_k > \mu_m$.

Heuristic 6 *Give processor k priority over processor m, if* $\xi_k < \xi_m$.

Heuristic 7 *Give processor k priority over processor m, if* $\eta_k > \eta_m$.

In the absence of theoretical results with general validity, the only way of evaluating all these heuristics is by experimentation. Some empirical results are reported in the next section.

6.5 Evaluation of the heuristics

Several experiments are conducted in order to evaluate these heuristics. The first of them, i.e. Experiment 1, involves a 100-processor parameterized system where the breakdown rates are about an order of magnitude smaller than the repair rates, which are in turn a couple of orders of magnitude smaller than the service rates. The experiment consists of generating 100 example systems or sample systems at random, i.e. random sets of values for the parameters μ_k, ξ_k and η_k ($k = 1, 2, \ldots, 100$). The latter are uniformly distributed within prescribed ranges: $\mu_k \in (1000, 100000)$, $\xi_k \in (0.01, 0.4)$ and $\eta_k \in (1.0, 39.0)$. For each example or sample system, the processing capacities under all the seven heuristics of the previous section, and also under the First-Come-First-Served repair strategy, are estimated by simulation (exact solutions are not feasible, due to the large state spaces of the Markov chains).

The above is repeated in Experiments 2 and 3, with different ranges for η_k, keeping the other parameter ranges the same as in Experiment 1. The results of these three experiments are shown in Figures 6.3 and 6.4.

The bar chart in Figure 6.3 shows, for each heuristic, the number of sample systems in which it produced the highest processing capacity among the eight heuristics (Heuristic 8 is the FCFS strategy). We observe that Heuristic 1 is most frequently best, while Heuristics 5, 6 and 8 are never best. However, the dominance of Heuristic 1 diminishes when the mean and variance of η_k increase (in Experiments 2 and 3), and hence the average utilisation of the repairman decreases. A somewhat surprising feature of this figure is the poor performance of Heuristic 3, and the relatively good performance of Heuristic 4.

It is interesting to note the improvement in processing capacity achieved by each heuristic, compared to FCFS. This is shown in Figure 6.4, where each bar measures the quantity $(\gamma_{heuristic} - \gamma_{FCFS})/\gamma_{FCFS}$, averaged over the 100 sample systems. It is rather remarkable that, with the exception of 5 and 6, all heuristics achieve roughly the same average relative improvement. More predictable is the fact that, the heavier the load on the repairman, the greater the magnitude of that improvement (however, even in the Experiment 3, a 25% average improvement is achieved by most heuristics).

Another strategy that can be used as a standard of comparison instead of FCFS

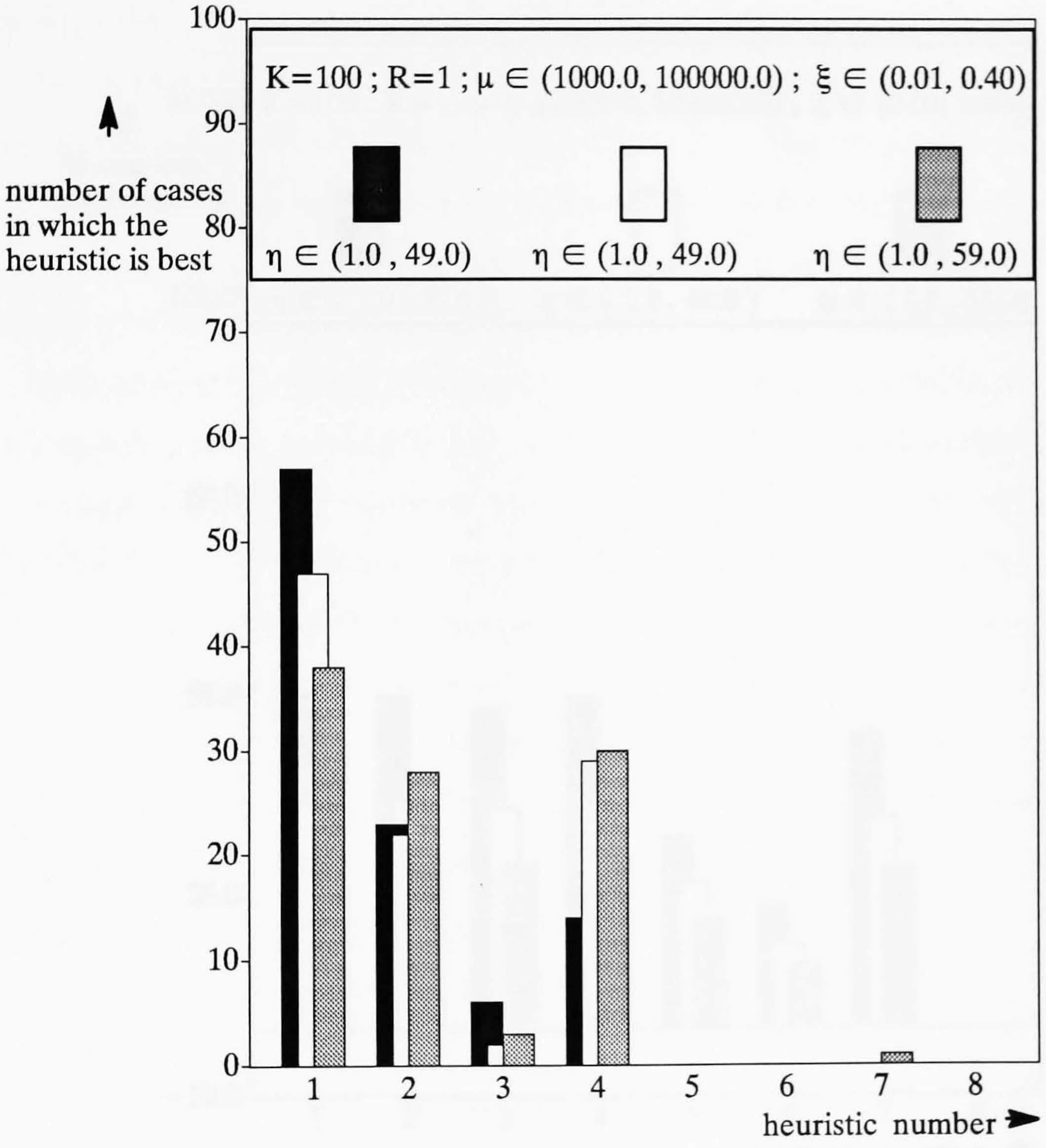

Figure 6.3: Performance of the heuristics in experiment with large K: number of cases in which each heuristic performed best.

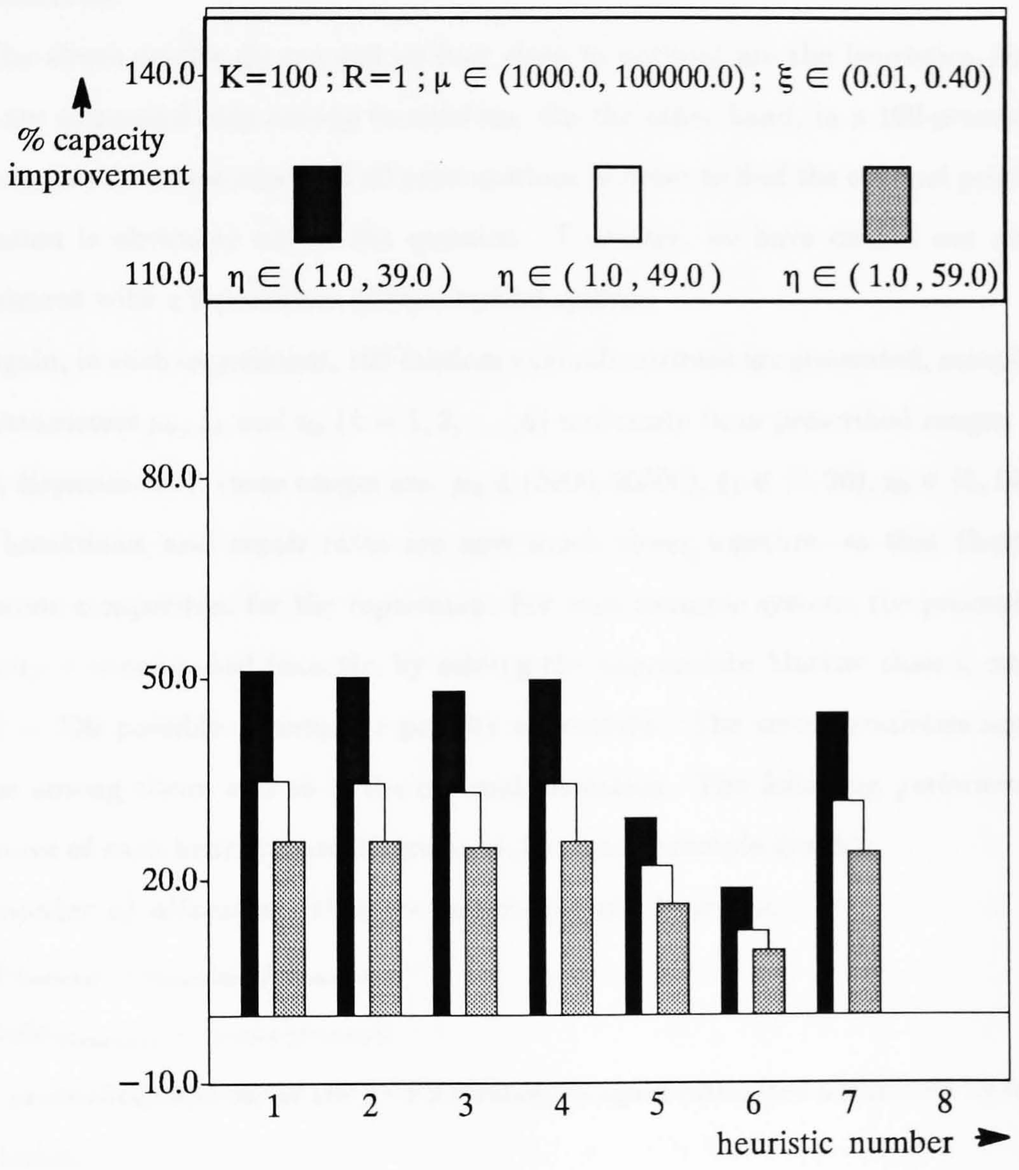

Figure 6.4: Performance of the heuristics in experiment with large K: improvements of capacity compared with FCFS repair policy.

is the random repair strategy, whereby at every scheduling decision instant, every one of the broken processors is equally likely to be selected for repair. This was tried, but no appreciable difference between the random and the FCFS strategies was observed.

The above results do not tell us how close to optimal are the heuristics, since they are compared only among themselves. On the other hand, in a 100-processor system, an exhaustive search of all permutations in order to find the optimal priority allocation is obviously out of the question. Therefore, we have carried out some experiment with a 6-processor parameterized system.

Again, in each experiment, 100 random example systems are generated, sampling the parameters μ_k, ξ_k and η_k $(k = 1, 2, \ldots, 6)$ uniformly from prescribed ranges.

In Experiment 4, those ranges are: $\mu_k \in (5000, 95000)$, $\xi_k \in (0, 30)$, $\eta_k \in (5, 145)$. The breakdown and repair rates are now much closer together, so that there is still some competition for the repairman. For each example system, the processing capacity γ is computed (exactly, by solving the appropriate Markov chain), under all $6! = 720$ possible preemptive priority allocations. The seven heuristics are of course among them, and so is the optimal allocation. The following performance measures of each heuristic are determined, for every example system:

$b =$ *number of allocations that are better than the heuristic*

$h = (\gamma_{optimal} - \gamma_{heuristic})/\gamma_{optimal}$

$a = 100(\gamma_{heuristic} - \gamma_{FCFS})/\gamma_{FCFS}$

(The processing capacity of the FCFS strategy is again estimated accurately by long simulation.)

A rough comparison between the heuristics is displayed in Table 6.1, where the quantities b, h and a are averaged over these 100 example systems.

The table shows that in this experiment the first three heuristics perform considerably better than the others. Their average relative distances from the optimum, h,

Table 6.1: Average performance of the heuristics in Experiment 4.

heuristic	average b	average h	average a
1	13.26	0.0027	33.0
2	1.44	0.0006	33.2
3	3.08	0.0018	33.0
4	22.28	0.0095	32.1
5	155.40	0.0891	19.3
6	247.23	0.1270	13.0
7	94.73	0.0368	28.2

are very small indeed. However, even the lease performing two heuristics, 5 and 6, yield processing capacities within about 12% of the optimum. The average relative improvement in capacity with respect to the FCFS strategy is considerable: it is on the order of 30% for all heuristics except 5 and 6.

The frequency histograms for b and h are displayed in Figures 6.5 and 6.6, respectively. Here, we have considered only the first three heuristics, for finer comparison. It can be seen that Heuristic 2 finds the best allocation in about 63% of the example systems, Heuristic 3 is optimal in about 45% of the example systems, and heuristic 1 is optimal in less than 30% of the example systems. Figure 6.6 demonstrates very clearly that, even when the best allocation is not found, the processing capacity achieved by all three heuristics is very close to optimal. In almost 90% of the example systems for Heuristic 2, 75% for Heuristic 3 and 50% for Heuristic 1, the relative error as measured by h is less than 0.001.

A similar experiment, i.e. Experiment 5, with a more heavily loaded repairman produced similar results, except that there the order of the first three heuristics was reversed: 1 was slightly better than 3, which was slightly better than 2. Again, the

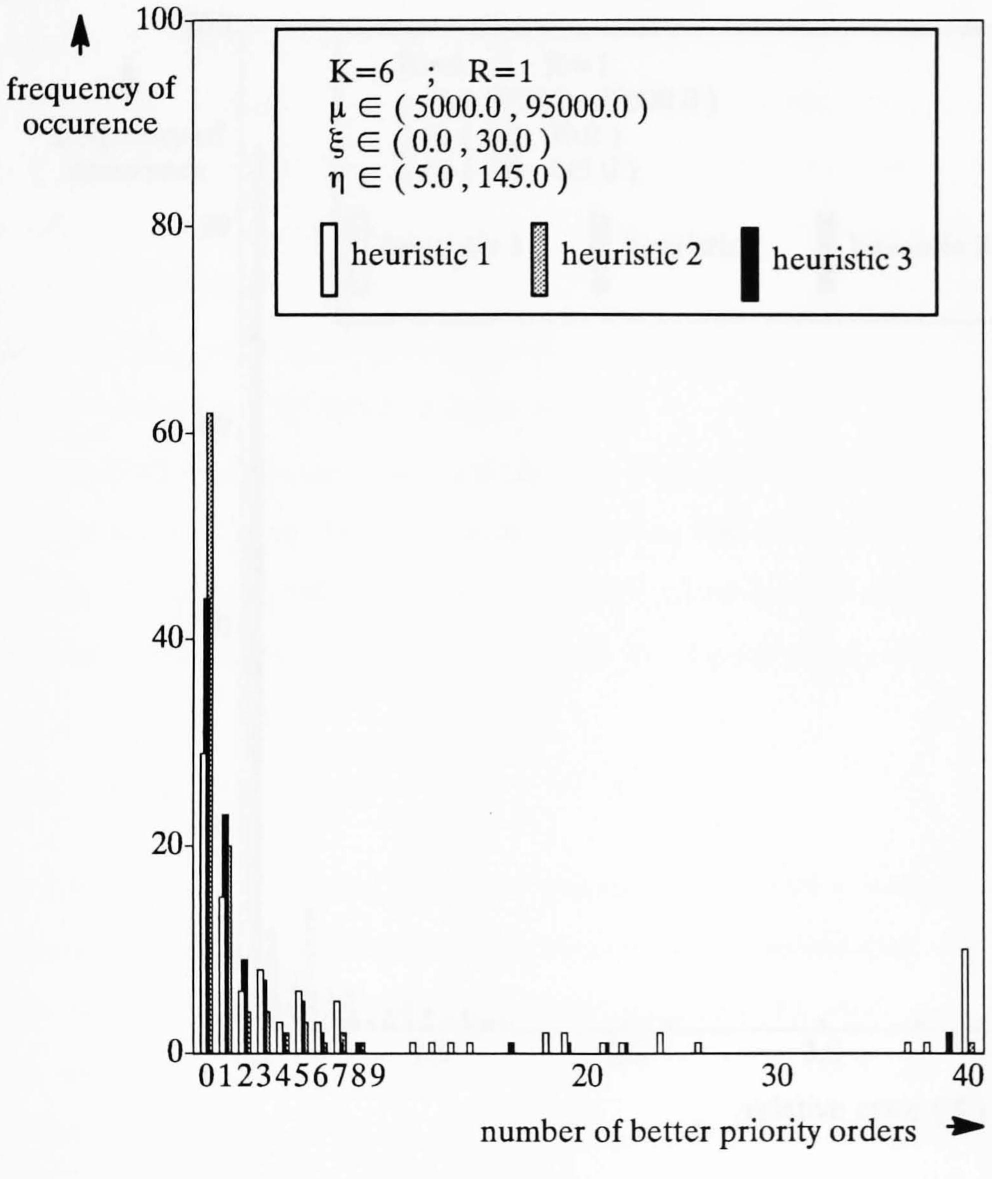

Figure 6.5: Performance of the heuristics 1-3 in Experiment 4, with $K = 6$: number of better allocations.

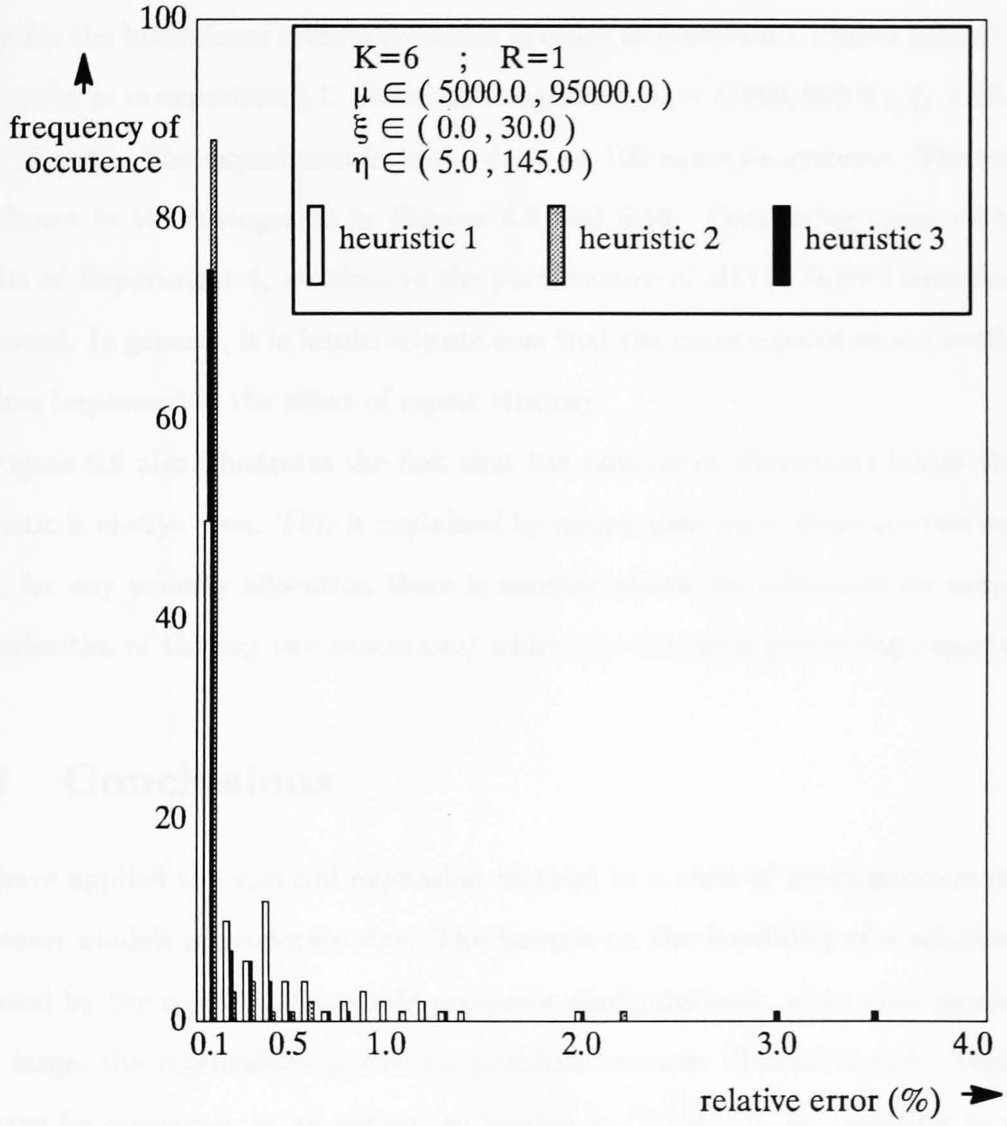

Figure 6.6: Performance of the heuristics 1-3 in Experiment 4, with $K = 6$: relative errors.

relative errors of all heuristics apart from 5 and 6 were very small. The results of this experiment are shown in Figures 6.7 and 6.8.

In Experiment 6, the number of repairmen is increased to 2. The parameter ranges for the service and repair rates are the same in Experiment 4, while the range for the breakdown rates is doubled, in order to maintain a similar load of each repairman as in experiment 1. Here, the ranges are: $\mu_k \in (5000, 95000)$, $\xi_k \in (0, 60)$, $\eta_k \in (5, 145)$. The experiment is again done on 100 example systems. The results are shown in the histograms in Figures 6.9 and 6.10. Comparing these with the results of Experiment 4, we observe the performance of all the three heuristics has improved. In general, it is intuitively obvious that the more repairmen are available, the less important is the effect of repair strategy.

Figure 6.9 also illustrates the fact that the number of allocations better than a heuristic is always even. This is explained by noting that when there are two repairmen, for any priority allocation there is another allocation (obtained by swapping the priorities of the top two processors) which has the same processing capacity.

6.6 Conclusions

We have applied the spectral expansion method to a class of heterogeneous multiprocessor models of moderate size. The bounds on the feasibility of a solution are dictated by the number of possible processor configurations: when that number is very large, the eigenvalue-eigenvector problem becomes ill-conditioned. This can perhaps be overcome, to an extent, as briefed in Chapter 5, by resorting to more accurate and robust numerical routines than the ones we have used.

As far as the optimal repair strategies are concerned, it is clear that any one among four or five heuristics would produce acceptable results in practice. In all cases, the priorities allocated may be preemptive or non-preemptive. It has been

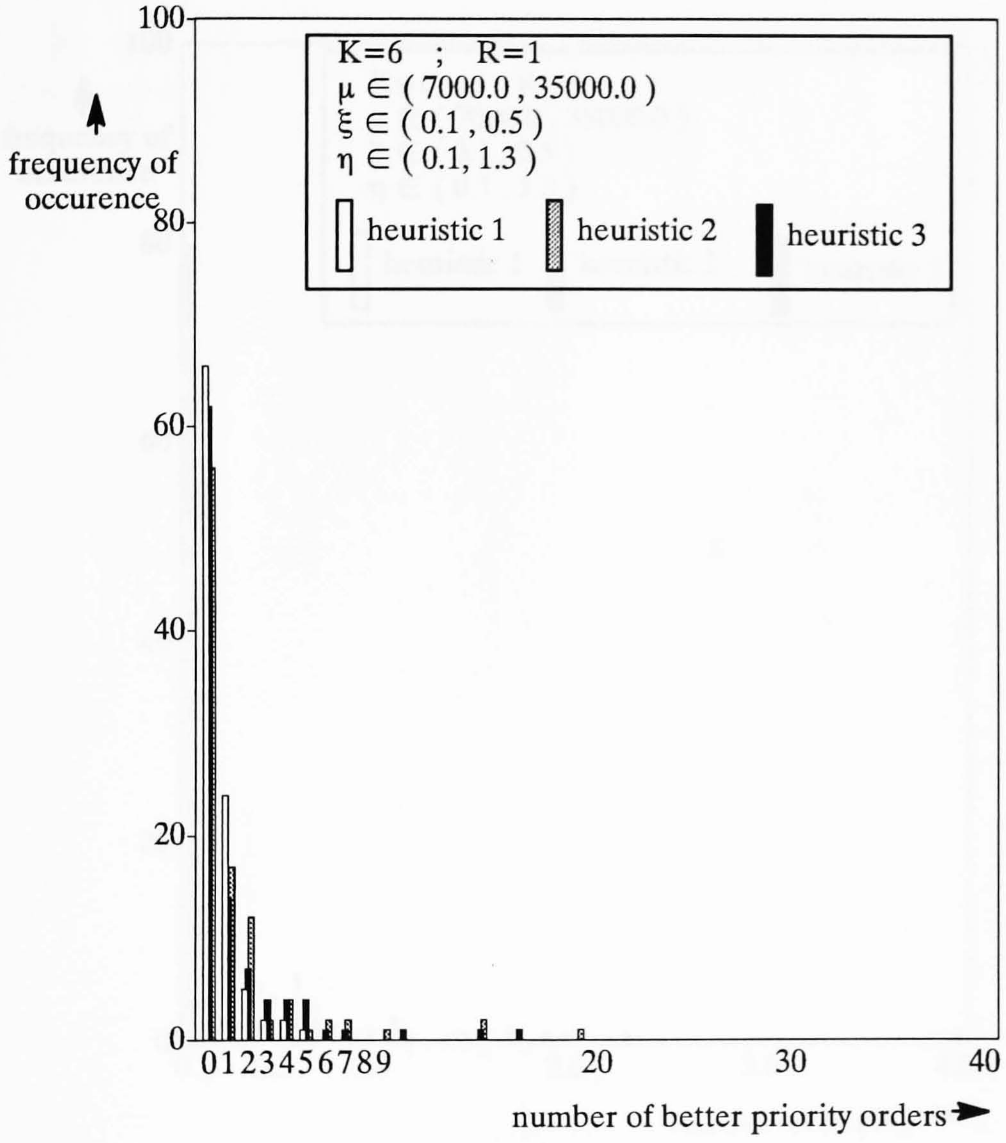

Figure 6.7: Performance of the heuristics 1-3 in Experiment 5, with $K = 6$: number of better allocations.

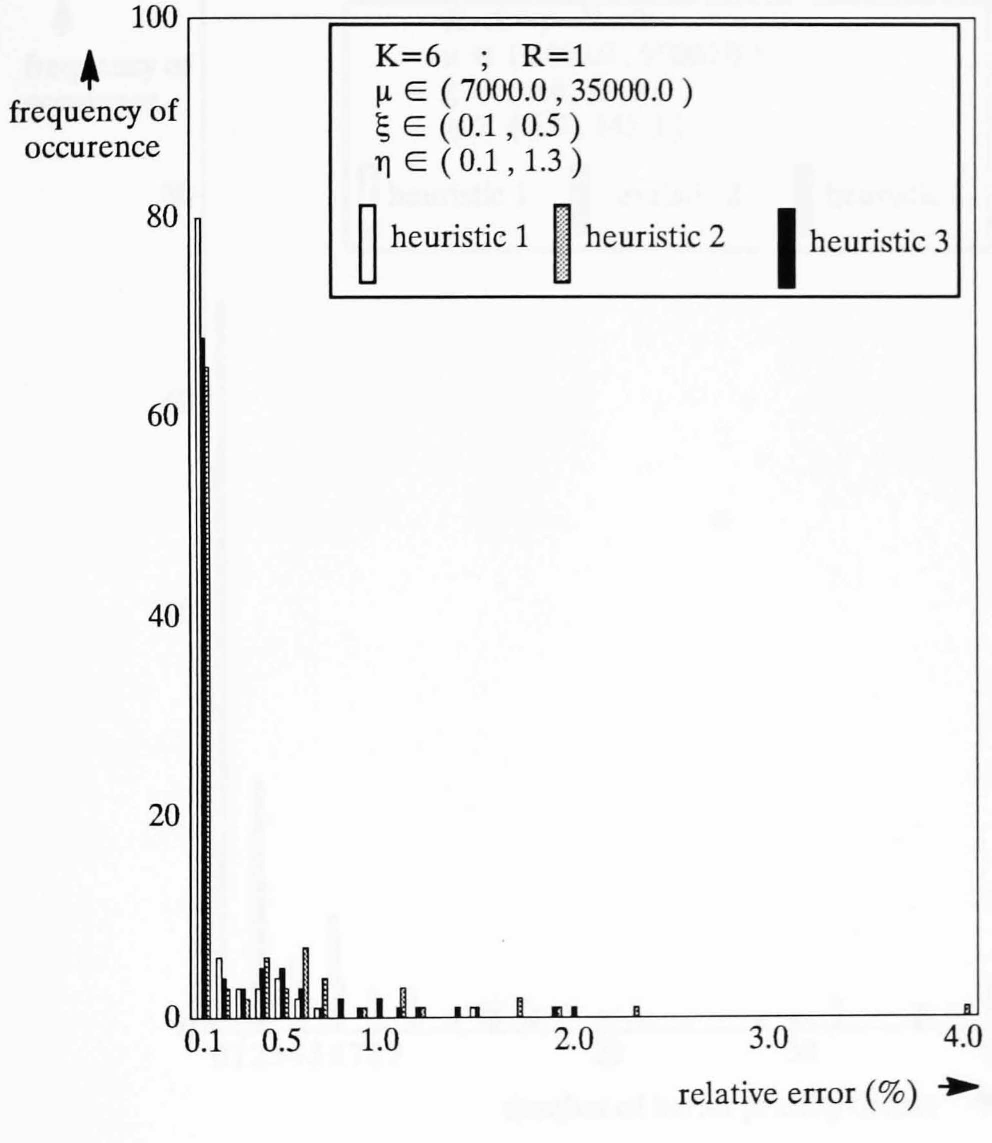

Figure 6.8: Performance of the heuristics 1-3 in Experiment 5, with $K = 6$: relative errors.

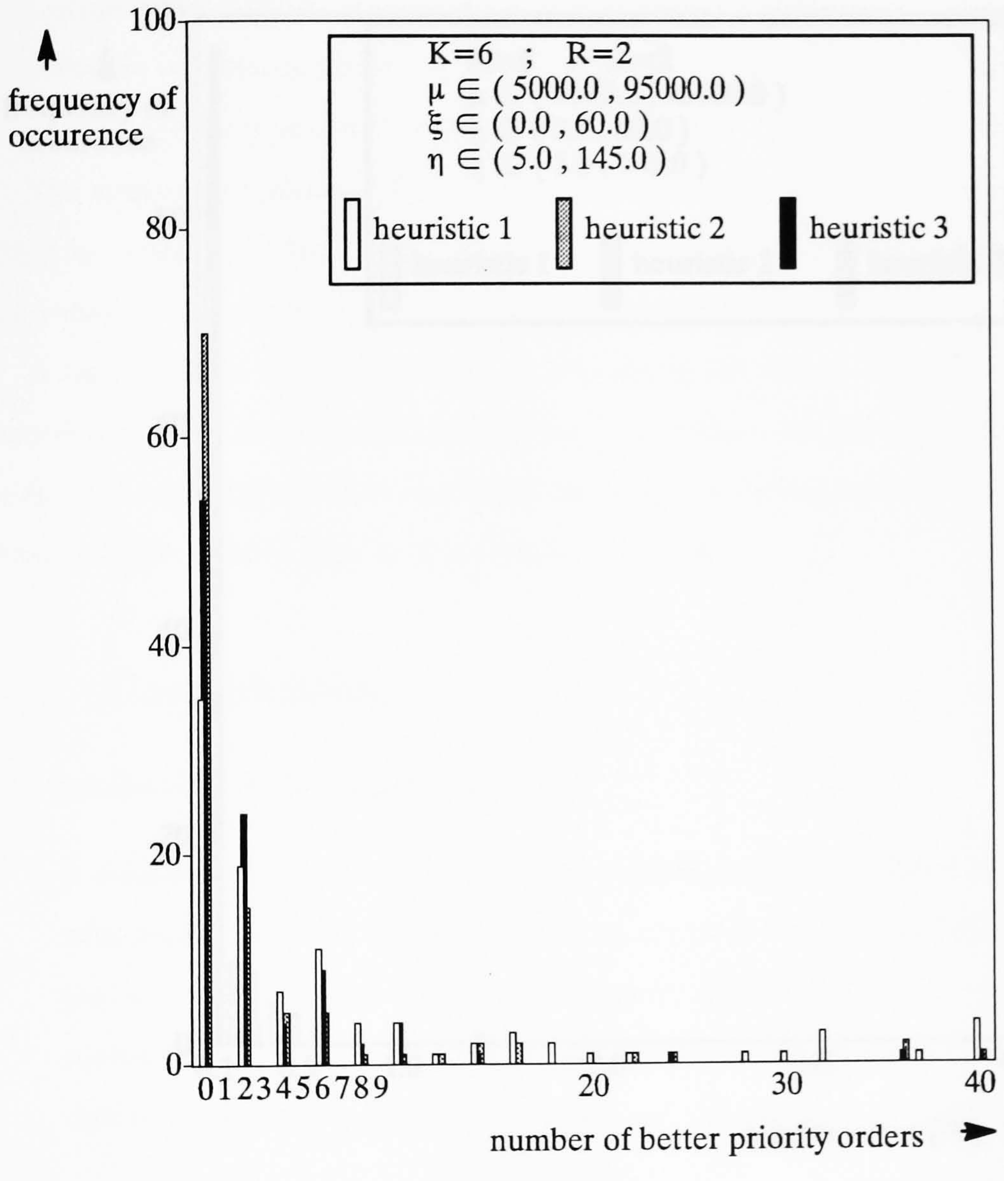

Figure 6.9: Performance of the heuristics 1-3 in Experiment 6, with $K = 6$: number of better allocations.

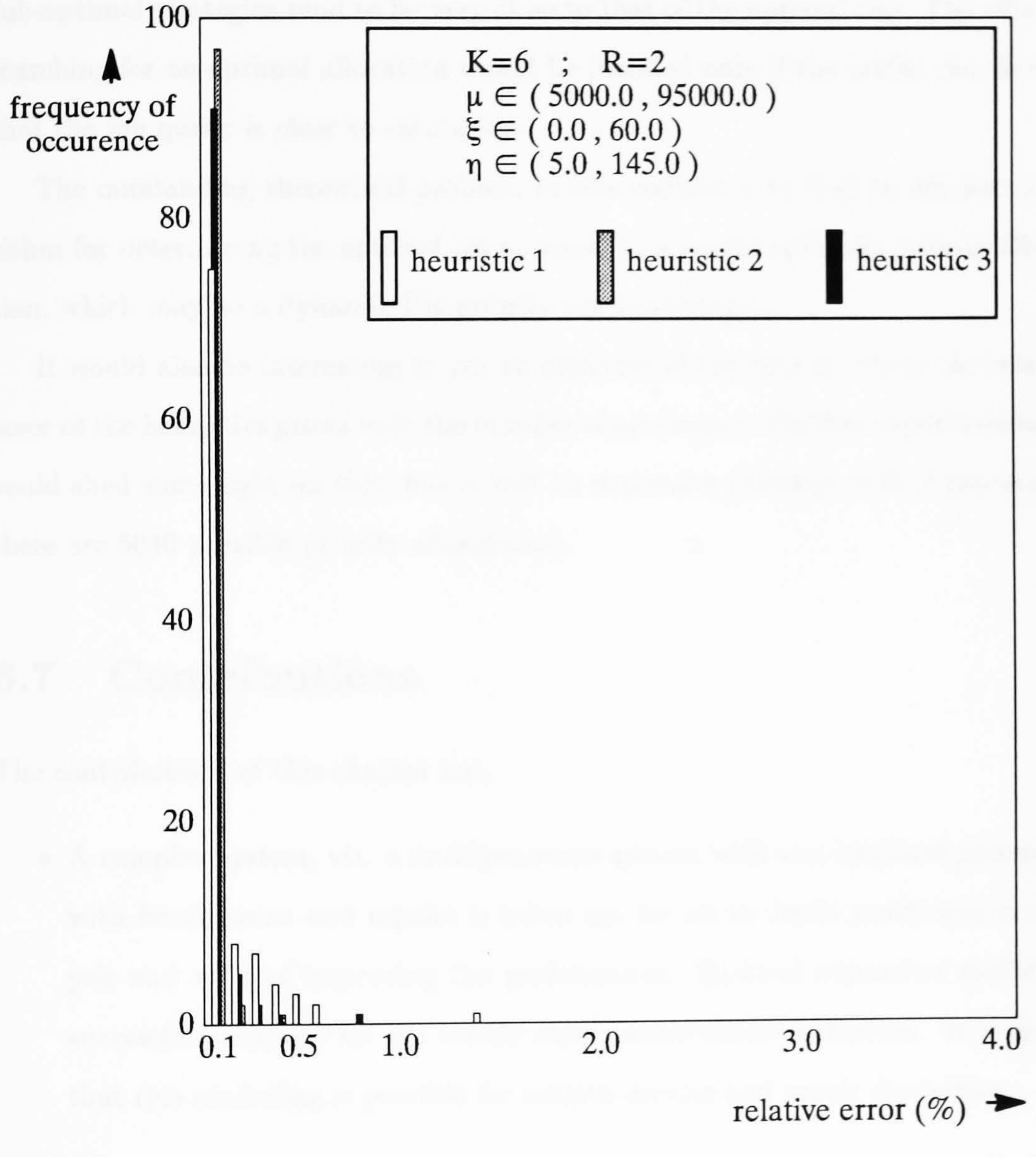

Figure 6.10: Performance of the heuristics 1-3 in Experiment 6, with $K = 6$: relative errors.

proved very recently, by a simple numerical counter example [36], that the optimal repair strategy is not always one of the static PR-priority repair strategies. However, our experiments provide a strong indication that the processing capacities of several sub-optimal strategies tend to be very close to that of the optimal one. The effort of searching for an optimal allocation would be justified only if the traffic rate is such that the job queue is close to saturation.

The outstanding theoretical problem in this context is to find an efficient algorithm for determining the optimal (as opposed to a nearly optimal) priority allocation, which may be a dynamic PR-priority repair strategy..

It would also be interesting to get an estimate of the rate at which the relative error of the heuristics grows with the number of processors. Further experimentation could shed some light on this, but it will be expensive (already with 7 processors, there are 5040 possible priority allocations).

6.7 Contributions

The contributions of this chapter are,

- A complex system, viz. a multiprocessor system with non-identical processors with breakdowns and repairs is taken up, for an in-depth performance analysis and ways of improving the performance. Spectral expansion method is successfully applied for the steady state performance evaluation. It is shown that this modelling is possible for various service and repair disciplines.

- It is illustrated how important the repair strategy is, especially when the contention for repair is high. The optimal repair strategy is obtained under some conditions. In the general case, several useful and nearly optimal heuristics for repair strategy are suggested. These strategies are evaluated against the

FCFS strategy, and against the optimal one in the case of multiprocessors of moderate size.

These research contributions are presented in our papers [6, 7].

Chapter 7

Queues with Breakdowns and Markov-modulated Arrivals

7.1 Introduction

In the earlier chapters, the arrivals considered have all been Poisson processes. But, in many practical systems, the arrival rates may vary randomly over time, quite sharply some times. Examples include the overflow from a finite trunk group [41], the superposition of packetized voice processes, and packet data [26, 28], in communications modelling. The Markov-modulated Poisson process (MMPP) has been extensively and effectively used for modelling such arrival processes. The MMPP is a non-renewal, doubly stochastic process where the Poisson arrival rate is varied according to the state of an independent continuous time Markov process. The MMPP is a Markov-renewal process, and it qualitatively models the time-varying arrival rates and captures some of the important correlations between the inter-arrival times while still remaining analytically tractable. An excellent review of the literature on the use of MMPP, and some useful results on MMPP's are found in [16].

Spectral expansion can solve certain queueing problems involving MMPP arrivals. That is the subject of this chapter. Such queues occur frequently in communication systems and other practical systems. We solve a MMPP/M/1 queue with unbounded queueing capacity, and with server breakdowns and repairs, for steady-state performance measures. Apart from this, we also approximately model the departure process from that queue, as an interrupted Poisson process (IPP) and solve for its parameters.

Section 7.2 describes the queue to be analysed. It also describes the spectral expansion modelling of that queue, followed by some numerical results on an example system. Section 7.3 is on modelling the departure process, approximately, as an IPP. It also presents results on the validation of that model. Section 7.4 concludes this chapter.

7.2 The MMPP/M/1 queue with breakdowns and repairs

We are interested in the analysis of a single server queue with MMPP job-arrivals. The server is prone to breakdowns and the broken server is set into repair process immediately after it is broken. The server becomes operative again following the completion of its repair. The service, breakdown and repair times of the server are exponentially distributed, with rates μ , ξ , and η respectively. When operative the server serves jobs , one at a time. When a job's service is interrupted because of a server breakdown, its service is resumed when the server is available, on the basis of *resume* or *repeat with re-sampling* discipline.

Let, T , be the number of phases of the MMPP arrival process, numbered as, $0, 1, \ldots, T-1$. That means, the associated Markov process has T states. Let Θ be the generator matrix of the associated continuous time, irreducible, homogeneous

Markov process, of the MMPP. Θ is of size $T \times T$. $\Theta(i,k) = \theta_{i,k}$, $0 \le i,k \le T-1$ and $i \ne k$, is the transition rate from phase i to phase k of the MMPP. $\Theta(i,i) = -\theta_i$ is given by,

$$\Theta(i,i) = -\sum_{k=0\,\&\,k\neq i}^{T-1} \Theta(i,k)\ ;\ i = 0,1,\ldots,T-1\ .$$

In other words,

$$\theta_i = \sum_{k=0\,\&\,k\neq i}^{T-1} \theta_{i,k}\ ;\ i = 0,1,\ldots,T-1\ .$$

Let σ_i be the arrival rate of the MMPP in phase i. Let Σ be the diagonal matrix, of size $T \times T$, whose i^{th} diagonal element is σ_i. Clearly, the MMPP process is characterized by the square matrices Θ and Σ.

Let c_i, $i = 0,1,\ldots,T-1$, be the steady state probability that the MMPP is in phase i, at any given time. Then c_i's can be computed using the following equations:

$$(c_0, c_1, \ldots, c_{T-1})\Theta = 0\ ; \qquad (c_0, c_1, \ldots, c_{T-1})\mathrm{e} = 1\ . \tag{7.1}$$

Hence, the total average arrival rate, $\widehat{\sigma}$, of jobs to the queue is,

$$\widehat{\sigma} = \sum_{i=0}^{T-1} c_i\sigma_i\ . \tag{7.2}$$

The effective average service rate of the server, taking into consideration the breakdowns and repairs, is $\widehat{\mu}$. This is given by,

$$\widehat{\mu} = \mu \cdot \frac{\eta}{\eta + \xi}\ . \tag{7.3}$$

It is then intuitively clear that the system is ergodic if, and only if $\widehat{\mu} > \widehat{\sigma}$. Regarding rigorous proof of this statement, the same remark as on page 101 applies here. We assume this condition holds good, for our steady state analysis.

The state of the system, at any time t, can be specified completely by three discrete random variables. They are, (i) the phase of the arrival process, $\Phi(t)$, an integer random variable varying from 0 to $T-1$, (ii) whether the server is broken or operative, a boolean random variable, $O(t)$, *broken* or *operative*, (iii) the number of jobs in the system including the one in service, if any, $J(t)$, unbounded integer random variable greater than or equal to 0. (i) and (ii) are of finite range, they can be combined into a single integer random variable $I(t)$, $I(t) \in (0, 1, \ldots, 2T-1)$, by appropriately defining $I(t)$. A possible definition of $I(t)$, the one chosen in this analysis, is as follows:

The first T states of $I(t)$, given by $I(t) = 0, 1, \ldots, T-1$, represent the breakdown states of the server, with respective arrival phases of the MMPP. The remaining T states, numbered $T, T+1, \ldots, 2T-1$, represent all the operative states of the server, along with arrival phases $0, 1, \ldots, T-1$ of the MMPP, respectively.

Now, we can represent the state of the system by two integer valued random variables, $(I(t), J(t))$, $J(t) \in (0, 1, \ldots)$, where $J(t)$ is the random variable (iii) mentioned above, representing the number of jobs in the system at time t.

This system can now be modelled by the Markov process X, as we shall see by its parameters.

$I(t) = 0, 1, \ldots, N$, $N = 2T-1$, are the $N+1$ *operative states* of the model or system. Note that, an operative state of the system is different from the operative state of the server. A breakdown or a repair of the server changes the value $I(t)$, but does not change the values of $J(t)$. An arrival of a new job or a departure of a job after service, changes $J(t)$, but does not change the random variable $I(t)$.

The purely lateral transition rate matrices, A and A_j, of size $(N+1) \times (N+1)$, are given by,

$$\begin{aligned}
A_j &= A \; ; \; j \geq 0 \, , \\
A(i, T+i) &= \eta \; ; \; i = 0, 1, \ldots, T-1 \, , \\
A(T+i, i) &= \xi \; ; \; i = 0, 1, \ldots, T-1 \, , \\
A(i, k) &= \theta_{i,k} \; ; \; i, k = 0, 1, \ldots, T-1 \; ; \; i \neq k \, , \\
A(T+i, T+k) &= \theta_{i,k} \; ; \; i, k = 0, 1, \ldots, T-1 \; ; \; i \neq k \, , \qquad (7.4)
\end{aligned}$$

all other elements of A are zeroes. The one-step upward transition rate matrices, B and B_j, of size $(N+1) \times (N+1)$, are diagonal matrices. They are,

$$\begin{aligned} B_j &= B \; ; \; j \geq 0 \, , \\ B(i,i) &= \sigma_i \; ; \; i = 0,1,\ldots,T-1 \, , \\ B(T+i,T+i) &= \sigma_i \; ; \; i = 0,1,\ldots,T-1 \, , \\ B(i,k) &= 0 \; ; \; i \neq k \, . \end{aligned} \tag{7.5}$$

C and C_j are the one-step downward transition rate matrices. Since, the departures do not change $I(t)$, unless such a change occurs by an independent coincidence, C and C_j are diagonal matrices. They are given by,

$$\begin{aligned} C_0 &= 0 \, , \\ C_j &= C \; ; \; j \geq 1 \, , \\ C(i,i) &= 0 \; ; \; i = 0,1,\ldots,T-1 \, , \\ C(T+i,T+i) &= \mu \; ; \; i = 0,1,\ldots,T-1 \, , \\ C(i,k) &= 0 \; ; \; i \neq k \, . \end{aligned} \tag{7.6}$$

Clearly, the threshold M is given by $M = 1$. Hence, the spectral expansion solution for the steady state probability $p_{i,j}$ and the row vectors $\mathbf{v}_j$ are valid for $j \geq 0$. $p_{i,j}$ and $\mathbf{v}_j$ are defined as,

$$\begin{aligned} p_{i,j} &= \lim_{t \to \infty} P(I(t) = i, J(t) = j) \; ; \; i = 0,1,\ldots,N \; ; \; j = 0,1,\ldots \\ \mathbf{v}_j &= (p_{0,j}, \, p_{1,j}, \ldots, \, p_{N,j}) \; ; \; j = 0,1,\ldots \end{aligned}$$

From this distribution $\{p_{i,j}\}$, it is possible to find the steady state probability of the state of the system, as,

$$\begin{aligned} \lim_{t \to \infty} P(\, \Phi(t) = i, O(t) = broken, J(t) = j\,) &= p_{i,j} \, , \\ \lim_{t \to \infty} P(\, \Phi(t) = i, O(t) = operative, J(t) = j\,) &= p_{T+i,j} \, , \end{aligned} \tag{7.7}$$

The steady state performance measures, such as moments of queuelengths, server utilization, can be computed from this probability distribution.

7.2.1 An example

This section is intended to illustrate the system using a simple example. The example considered, has a 3-phase MMPP arrival process. The phases are numbered as, $0, 1, 2$. Phase 1 is the normal phase of the arrival process, with arrival rate σ. Phases 0 and 2 are abnormal phases of the arrival process. In phase 2, the arrivals are very high, 10σ, 10 times the normal arrival rate. In phase 0, there is a highly diminished arrival rate given by, 0.1σ. The transitions rates are, $\theta_{1,2} = 0.003\theta$, $\theta_{1,0} = 0.007\theta$, $\theta_{2,1} = 0.05\theta$, $\theta_{0,1} = 0.1\theta$, $\theta_{2,0} = 0.05\theta$ and $\theta_{0,2} = 0.1\theta$. Thus, σ and θ are the variable parameters of the system.

In a transition to phase 1, the average time the arrival process spends in that phase, before a subsequent transition, is $\frac{100}{\theta}$. Similarly, those times for phase 0 and 2 are, $\frac{5}{\theta}$ and $\frac{10}{\theta}$, respectively. Hence, θ regulates the actual duration of the phases, though their relative durations do not depend on θ.

Using the equations (7.1, 7.2), the total average arrival rate is computed as, $\widehat{\sigma} = 1.63761\sigma$. Notice that $\widehat{\sigma}$ also does not depend on θ.

Let the parameters of the server be, $\mu = 25.0$, $\xi = 0.01$, $\eta = 0.1$. Hence, $\widehat{\mu} = 22.727273$. The system is ergdic if $\widehat{\sigma} < 22.727273$, or equivalently, if $\sigma < 13.8783$.

For $\theta = 1.0$, the mean queuelength, $E(J)$, is computed for various values of σ. These results are plotted in Figure 7.1, $E(J)$ vs. σ. These results are compared with the queuelengths of a simplified model, an M/M/1 queue with breakdowns and repairs. The bursty arrival rates are smoothed to a Poisson, in this M/M/1 model, that is with uniform arrival rate equal to $\widehat{\sigma} = 1.63761\sigma$.

The results are quite clear. The M/M/1 results are gross underestimation of the original model, MMPP/M/1 queue. Hence, it can be concluded that the burstiness

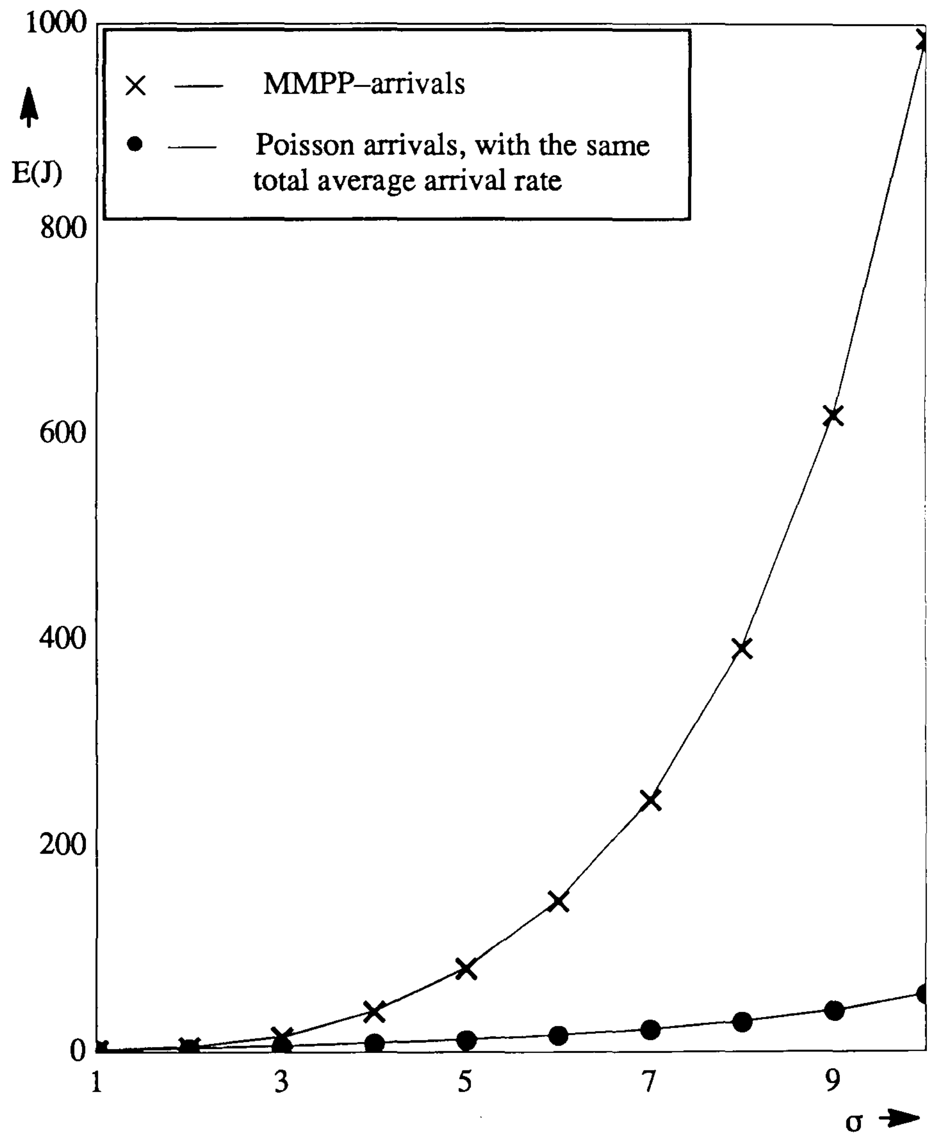

Figure 7.1: Mean queuelength vs. σ, in the example.

of the arrivals affect the performance measures greatly. Mere approximation by a Poisson process, could give considerably incorrect results. Hence, the importance and necessity of the models considered in this chapter.

7.3 Departure process modelling

The system and the model analysed in section 7.2 occur sometimes as stand-alone, but in many practical situations, as coupled with some other systems. For example, in assembly lines, the departed jobs from one such MMPP/M/1 queue are often split into several flows, for different operations, or combined with some other flows. In networks of queues, the departed jobs may be fed to other relevant queues, as arrivals, for further servicing. Hence, there is a case for the analysis of the departure process of the system in section 7.2, and develop an exact or approximate, analytically tractable model for it.

The most appropriate model for the departure process, of course, would be another MMPP. But, since we are interested in a simplified model, we have chosen the interrupted Poisson process (IPP) as the model. The IPP is essentially an MMPP with two phases, and the flow rate in one of those phases is equal to 0. Let the parameters of this departure process-IPP be (ν, α, β). That means: the IPP has two phases, Phase 1 and Phase 2. In Phase 1, the departure rate is ν. In Phase 2, the departure rate is 0. The transition rate from Phase 1 to Phase 2 is α, and that from Phase 2 to Phase 1 is β. The question before us is how to determine, appropriately, the values of these parameters. This is done by computing the moments of the inter-departure times, from the parameters of the system, and working out the IPP parameters from these computed moments.

Consider the system in steady state, and at a time t_1. Exactly at time t_1, there may or may not be a departure. Let the next or subsequent departure, strictly after

t_1, be at time t_2 ($t_2 > t_1$). Let $\tau = t_2 - t_1$ be the random variable, representing the interval between t_1 and t_2. The departure process would be stationary when the system, or the process X, is in steady state. Let $\Delta(s)$ be the Laplace transform of the density of inter-departure interval. If there is a departure at time t_1, then $O(t_1) = \textit{operative}$ and τ would represent the inter-departure interval. In order to derive an expression for $\Delta(s)$, let the following Laplace transforms be defined. They are,

$\Upsilon(s)$ – Laplace transform of the density of the random variable τ, if $O(t_1) = \textit{broken}$ and $J(t_1) \geq 1$.

$\Omega(s)$ – Laplace transform of the density of the random variable τ, if $O(t_1) = \textit{operative}$ and $J(t_1) \geq 1$. $\Omega(s)$ would be the same whether or not there is a departure exactly at t_1.

$\Gamma_i(s)$ – Laplace transform of the density of the random variable τ, if the state of the system at t_1 is, $I(t_1) = i$ and $J(t_1) = 0$. $\Gamma_i(s)$ ($i = T, T+1, \ldots, N$) would be the same whether or not there is a departure exactly at t_1.

It is possible to express $\Upsilon(s)$ and $\Omega(s)$ in terms of the parameters of the system, as follows. At t_1, if $O(t_1) = \textit{operative}$ and $J(t_1) \geq 1$, then the next departure may occur before a breakdown, or, the next breakdown may occur before a departure. The probabilities for these two events are $\frac{\mu}{\mu+\xi}$, $\frac{\xi}{\mu+\xi}$, respectively. If the latter occurs, then the state of the system becomes $O(t) = \textit{broken}$ and $J(t) \geq 1$, at the time of the breakdown. Hence, we have

$$\Omega(s) = \frac{\mu}{\mu+\xi} \cdot \frac{\mu+\xi}{\mu+\xi+s} + \frac{\xi}{\mu+\xi} \cdot \frac{\mu+\xi}{\mu+\xi+s} \cdot \Upsilon(s) . \qquad (7.8)$$

On the otherhand, if $O(t_1) = \textit{broken}$ and $J(t_1) \geq 1$, then the system reaches $O(t) = \textit{operative}$ and $J(t) \geq 1$, at a rate η. Only after this, a departure can take place. Hence, we can write,

$$\Upsilon(s) = \frac{\eta}{\eta + s} \cdot \Omega(s) \, . \tag{7.9}$$

From equations (7.8) and (7.9), the Laplace transforms, $\Upsilon(s)$ and $\Omega(s)$, can be expressed as,

$$\Upsilon(s) = \frac{\mu\eta}{(\mu + \xi + s)(\eta + s) - \eta\xi} \, ; \tag{7.10}$$

$$\Omega(s) = \frac{\mu(\eta + s)}{(\mu + \xi + s)(\eta + s) - \eta\xi} \, . \tag{7.11}$$

Using the similar next-event analysis, as done above, $\Gamma_i(s)$, $i = 0, 1 \dots, N$, can be derived as follows.

At t_1, if $I(t_1) = i$ $(i = 0, 1, \dots, T-1)$, $J(t_1) = 0$, then the next significant event of interest, before a departure, can be, (i) arrival of a new job making the state of the system $I(t) = i$, $J(t) = 1$, for some value of t, or, (ii) completion of repair of the server leading to $I(t) = T + i$ and $J(t) = 0$, or, (iii) a change of phase of the MMPP, from phase i to phase k $(k \neq i)$, leading to $I(t) = k$ $(k = 0, 1, \dots, T-1 \; ; \; k \neq i)$ and $J(t) = 0$. The probabilities of the events (i), (ii) and (iii) are $\frac{\sigma_i}{\sigma_i + \eta + \theta_i}$, $\frac{\eta}{\sigma_i + \eta + \theta_i}$ and $\frac{\theta_{i,k}}{\sigma_i + \eta + \theta_i}$ $(k = 0, 1, \dots, T-1 \; ; \; k \neq i)$, respectively. Hence, we have

$$\begin{aligned} \Gamma_i(s) \;=\; & \frac{\sigma_i}{\sigma_i + \eta + \theta_i} \cdot \frac{\sigma_i + \eta + \theta_i}{\sigma_i + \eta + \theta_i + s} \cdot \Upsilon(s) + \frac{\eta}{\sigma_i + \eta + \theta_i} \cdot \frac{\sigma_i + \eta + \theta_i}{\sigma_i + \eta + \theta_i + s} \cdot \Gamma_{T+i}(s) \\ & + \sum_{k=0 \,\&\, k \neq i}^{T-1} \frac{\theta_{i,k}}{\sigma_i + \eta + \theta_i} \cdot \frac{\sigma_i + \eta + \theta_i}{\sigma_i + \eta + \theta_i + s} \cdot \Gamma_k(s) \end{aligned} \tag{7.12}$$

In case, at t_1, if $I(t_1) = T + i$ $(i = 0, 1, \dots, T-1)$, $J(t_1) = 0$, then the next significant event of interest, before a departure, can be, (i) arrival of a new job making the state of the system $I(t) = T + i$, $J(t) = 1$, or, (ii) a breakdown of the server leading to $I(t) = i$ and $J(t) = 0$, or, (iii) a change of phase of the MMPP, from phase i to phase k $(k \neq i)$, leading to $I(t) = T + k$ $(k = 0, 1, \dots, T-1 \; ; \; k \neq i)$ and $J(t) = 0$. The probabilities of the events (i), (ii) and (iii) are $\frac{\sigma_i}{\sigma_i + \xi + \theta_i}$, $\frac{\xi}{\sigma_i + \xi + \theta_i}$ and $\frac{\theta_{i,k}}{\sigma_i + \xi + \theta_i}$ $(k = 0, 1, \dots, T-1 \; ; \; k \neq i)$, respectively. Hence, we can write

$$\begin{aligned}\Gamma_{T+i}(s) &= \frac{\sigma_i}{\sigma_i+\xi+\theta_i}\cdot\frac{\sigma_i+\xi+\theta_i}{\sigma_i+\xi+\theta_i+s}\cdot\Omega(s)+\frac{\xi}{\sigma_i+\xi+\theta_i}\cdot\frac{\sigma_i+\xi+\theta_i}{\sigma_i+\xi+\theta_i+s}\cdot\Gamma_i(s)\\ &+\sum_{k=0\,\&\,k\neq i}^{T-1}\frac{\theta_{i,k}}{\sigma_i+\xi+\theta_i}\cdot\frac{\sigma_i+\xi+\theta_i}{\sigma_i+\xi+\theta_i+s}\cdot\Gamma_{T+k}(s) \qquad (7.13)\end{aligned}$$

Equations (7.12) and (7.13) can be simplified as,

$$\Gamma_i(s)=\frac{\sigma_i\Upsilon(s)+\eta\Gamma_{T+i}(s)+\sum_{k=0\,\&\,k\neq i}^{T-1}\theta_{i,k}\Gamma_k(s)}{\sigma_i+\eta+\theta_i+s}\ ;\ i=0,1,\ldots,T-1 \qquad (7.14)$$

$$\Gamma_{T+i}(s)=\frac{\sigma_i\Omega(s)+\xi\Gamma_i(s)+\sum_{k=0\,\&\,k\neq i}^{T-1}\theta_{i,k}\Gamma_{T+k}(s)}{\sigma_i+\xi+\theta_i+s}\ ;\ i=0,1,\ldots,T-1 \qquad (7.15)$$

Equations (7.14) and (7.15) are $N+1$ linear simultaneous equations in $N+1$ unknowns, namely, $\Gamma_i(s)$'s. Hence, the latter can be found exactly, by solving these equations.

It is reasonable to say that at any departure epoch of a job, the server is in operative state. Now, let us assume, without loss of generality, there is a departure exactly at t_1. Rather, this is done simply by choosing t_1 at a departure epoch. Then τ represents the inter-departure time. The Laplace transform, $\Delta(s)$, of the density of this random variable τ, can now be written as,

$$\Delta(s)=\sum_{i=0}^{T-1}\pi_{i,0}\Gamma_{T+i}(s)+(1-\pi_0)\Omega(s)\ , \qquad (7.16)$$

where,

π_0 is the probability that $J(t)=0$ at a job departure epoch.

$\pi_{i,0}$ is the probability that $J(t)=0$ and the arrival phase $\Phi(t)=i$, at a job departure epoch. It is evident, $\sum_{i=0}^{T-1}\pi_{i,0}=\pi_0$

$\pi_{i,0}$'s are related to some of the steady state probabilities computed by spectral expansion, as,

$$\begin{aligned}\pi_{i,0} &= \frac{P(\Phi(t) = i, O(t) = operative, J(t) = 1) \cdot \mu}{\widehat{\sigma}} \\ &= \frac{p_{T+i,1} \cdot \mu}{\widehat{\sigma}} \; ; \; i = 0, 1, \ldots, T-1 \end{aligned} \tag{7.17}$$

The probabilities $p_{T+i,1}$ $(i = 0, 1, \ldots, T-1)$ can be computed exactly using spectral expansion solution. Using these results of the spectral expansion solution, $\pi_{i,0}$'s and π_0 can be computed exactly from (7.17).

7.3.1 Moments of the inter-departure interval

In order to fit an IPP for the departure process, what we need are the computed values of the first 3 moments of the random variable τ. Let $\Omega^{(k)}(s)$ and $\Upsilon^{(k)}(s)$ be the k^{th} derivatives of $\Omega(s)$ and $\Upsilon(s)$ respectively, with respect to s. Also, let $\Gamma_i^{(k)}(s)$ $(i = 0, 1, \ldots, N)$ be the k^{th} derivative of $\Gamma_i(s)$, with respect to s. Then, by differentiating equation (7.16) 3 times, we get,

$$\Delta^{(1)}(s) = \sum_{i=0}^{T-1} \pi_{i,0}\Gamma_{T+i}^{(1)}(s) + (1 - \pi_0)\Omega^{(1)}(s) \; ; \tag{7.18}$$

$$\Delta^{(2)}(s) = \sum_{i=0}^{T-1} \pi_{i,0}\Gamma_{T+i}^{(2)}(s) + (1 - \pi_0)\Omega^{(2)}(s) \; ; \tag{7.19}$$

$$\Delta^{(3)}(s) = \sum_{i=0}^{T-1} \pi_{i,0}\Gamma_{T+i}^{(3)}(s) + (1 - \pi_0)\Omega^{(3)}(s) \; ; \tag{7.20}$$

where $\Delta^{(k)}(s)$ is the k^{th} derivative of $\Delta(s)$ with respect to s. If d_1, d_2 and d_3 are the first 3 moments of τ, then we have,

$$d_1 = -\Delta^{(1)}(s)|_{s=0} \; ; \; d_2 = \Delta^{(2)}(s)|_{s=0} \; ; \; d_3 = -\Delta^{(3)}(s)|_{s=0} \, . \tag{7.21}$$

By differentiating equation (7.10), successively 3 times, and evaluating the derivatives $\Upsilon^{(1)}(s)$, $\Upsilon^{(2)}(s)$ and $\Upsilon^{(3)}(s)$, at $s = 0$, we get,

$$\Upsilon^{(1)}(0) = -\frac{(\eta + \mu + \xi)}{\eta\mu} \; ; \tag{7.22}$$

$$\Upsilon^{(2)}(0) = \frac{2[(\eta+\mu+\xi)^2 - \eta\mu]}{\eta^2\mu^2} ; \tag{7.23}$$

$$\Upsilon^{(3)}(0) = -\frac{6[(\eta+\mu+\xi)^3 - 2\eta\mu(\eta+\mu+\xi)]}{\eta^3\mu^3} ; \tag{7.24}$$

where, $\Upsilon^{(k)}(0) = \Upsilon^{(k)}(s)|_{s=0}$.

Similarly, by differentiating equation (7.11), successively 3 times, and evaluating the derivatives $\Omega^{(1)}(s)$, $\Omega^{(2)}(s)$ and $\Omega^{(3)}(s)$, at $s = 0$, we get

$$\Omega^{(1)}(0) = -\frac{(\eta+\xi)}{\eta\mu} ; \tag{7.25}$$

$$\Omega^{(2)}(0) = \frac{2[(\eta+\xi)^2 + \mu\xi]}{\eta^2\mu^2} ; \tag{7.26}$$

$$\Omega^{(3)}(0) = -\frac{6[(\eta+\xi)^3 + 2\eta\mu\xi + \mu^2\xi + 2\mu\xi^2]}{\eta^3\mu^3} \tag{7.27}$$

where, $\Omega^{(k)}(0) = \Omega^{(k)}(s)|_{s=0}$.

In order to compute d_1, d_2 and d_3, we also need to compute the values of $\Gamma_i^{(k)}(s)$, $k = 1,2,3$ and $0 \le i \le N$, at $s = 0$. Denote, $\Gamma_i^{(k)}(0) = \Gamma_i^{(k)}(s)|_{s=0}$. Differentiating the equations (7.14, 7.15), a total of $2 \cdot T$ equations with respect to s, and then substituting $s = 0$, we get the following equations.

$$\Gamma_i^{(1)}(0) = \frac{\sigma_i\Upsilon^{(1)}(0) + \eta\Gamma_{T+i}^{(1)}(0) - 1 + \sum_{k=0\,\&\,k\neq i}^{T-1}\theta_{i,k}\Gamma_k^{(1)}(0)}{\sigma_i + \eta + \theta_i} ; \; i = 0,1,\ldots,T-1 \tag{7.28}$$

$$\Gamma_{T+i}^{(1)}(0) = \frac{\sigma_i\Omega^{(1)}(0) + \xi\Gamma_i^{(1)}(0) - 1 + \sum_{k=0\,\&\,k\neq i}^{T-1}\theta_{i,k}\Gamma_{T+k}^{(1)}(0)}{\sigma_i + \xi + \theta_i} ; \; i = 0,1,\ldots,T-1 \tag{7.29}$$

Assuming $\Upsilon^{(1)}(0)$ and $\Omega^{(1)}(0)$ are known from the equations (7.22, 7.25), the above are $2{\cdot}T$ linear simultaneous equations in $2{\cdot}T$ unknowns, $\Gamma_i^{(1)}(0)$ ($i = 0,1,\ldots,2T-1$). Hence the values of these unknowns are obtained by solving the equations (7.28, 7.29).

Similar procedure is followed for obtaining the values of second and third derivatives, $\Gamma_i^{(2)}(0)$ and $\Gamma_i^{(3)}(0)$, for all the possible values of i. This is done as follows.

Differentiate equations (7.14) and (7.15), twice successively, with respect to s. Then, substitute $s = 0$. Then, we get,

$$\Gamma_i^{(2)}(0) = \frac{\sigma_i \Upsilon^{(2)}(0) + \eta \Gamma_{T+i}^{(2)}(0) - 2\Gamma_i^{(1)}(0) + \sum_{k=1 \,\&\, k\neq i}^{T-1} \theta_{i,k} \Gamma_k^{(2)}(0)}{\sigma_i + \eta + \theta_i} ;$$
$$i = 0, 1, \ldots, T-1 \qquad (7.30)$$

$$\Gamma_{T+i}^{(2)}(0) = \frac{\sigma_i \Omega^{(2)}(0) + \xi \Gamma_i^{(2)}(0) - 2\Gamma_{T+i}^{(1)}(0) + \sum_{k=1 \,\&\, k\neq i}^{T-1} \theta_{i,k} \Gamma_{T+k}^{(2)}(0)}{\sigma_i + \xi + \theta_i} ;$$
$$i = 0, 1, \ldots, T-1 \qquad (7.31)$$

Since $\Upsilon^{(2)}(0)$, $\Omega^{(2)}(0)$ and $\Gamma_i^{(1)}(0)$'s are known, equations (7.30) and (7.31) represent $2 \cdot T$ linear simultaneous equations in $2 \cdot T$ unknowns, the $\Gamma_i^{(2)}(0)$'s ($i = 0, 1, \ldots, 2T-1$). Hence, these unknowns can be computed by solving the linear simultaneous equations.

Third order differentiation of the equations (7.14) and (7.15), and substituting $s = 0$, leads to

$$\Gamma_i^{(3)}(0) = \frac{\sigma_i \Upsilon^{(3)}(0) + \eta \Gamma_{T+i}^{(3)}(0) - 3\Gamma_i^{(2)}(0) + \sum_{k=0 \,\&\, k\neq i}^{T-1} \theta_{i,k} \Gamma_i^{(3)}(0)}{\sigma_i + \eta + \theta_i} ;$$
$$i = 0, 1, \ldots, T-1 \qquad (7.32)$$

$$\Gamma_{T+i}^{(3)}(0) = \frac{\sigma_i \Omega^{(3)}(0) + \xi \Gamma_i^{(3)}(0) - 3\Gamma_{T+i}^{(2)}(0) + \sum_{k=0 \,\&\, k\neq i}^{T-1} \theta_{i,k} \Gamma_{T+i}^{(3)}(0)}{\sigma_i + \xi + \theta_i} ;$$
$$i = 0, 1, \ldots, T-1 \qquad (7.33)$$

Since $\Upsilon^{(3)}(0)$, $\Omega^{(3)}(0)$ and $\Gamma_i^{(2)}(0)$'s are already known, the above equations (7.32) and (7.33) are $2 \cdot T$ linear simultaneous equations in $2 \cdot T$ unknowns, $\Gamma_i^{(3)}(0)$'s ($i = 0, 1, \ldots, 2T-1$). Hence, the latter can be computed by solving these equations.

Denote, $\Delta^{(k)}(0) = \Delta^{(k)}(s)|_{s=0}$. Now, we are in a position to compute the first 3 moments of τ, as

$$d_1 = -\Delta^{(1)}(0) = -\sum_{i=0}^{T-1} \pi_{i,0} \Gamma_{T+i}^{(1)}(0) - (1 - \pi_0)\Omega^{(1)}(0) ;$$

$$\begin{aligned} d_2 &= \Delta^{(2)}(0) = \sum_{i=0}^{T-1} \pi_{i,0}\Gamma_{T+i}^{(2)}(0) + (1-\pi_0)\Omega^{(2)}(0) ; \\ d_3 &= -\Delta^{(3)}(0) = -\sum_{i=0}^{T-1} \pi_{i,0}\Gamma_{T+i}^{(3)}(0) - (1-\pi_0)\Omega^{(3)}(0) . \end{aligned}$$

It can be noted, $d_1 = 1/\widehat{\sigma}$, simply because the average rate of arrivals and average rate of departures are equal in steady state.

7.3.2 Fitting an IPP

Having obtained the first 3 moments of the random variable τ, the inter-departure time in the steady state, we now need to fit exactly or approximately a standard process for analytical modelling. Another MMPP would be ideal for this, yet we have chosen an IPP for simplicity at the cost of some accuracy. The IPP is a renewal process. ν, α and β are the parameters of this IPP, as described earlier. The idea now is to estimate these parameters from the computed moments of the inter-departure times , and other details.

The Laplace transform of the inter-departure/arrival time density, of this IPP is given by,

$$F(s) = \frac{\nu(\beta+s)}{\beta\nu + s(\alpha+\beta+\nu) + s^2} . \tag{7.34}$$

The first 3 moments of this IPP's inter-departure time, $-F^{(1)}(s)|_{s=0}$, $F^{(2)}(s)|_{s=0}$ and $-F^{(3)}(s)|_{s=0}$, respectively, are then given by,

$$\begin{aligned} -F^{(1)}(0) &= \frac{\alpha+\beta}{\beta\nu} ; \\ F^{(2)}(0) &= \frac{2[(\alpha+\beta)^2 + \alpha\nu]}{\beta^2\nu^2} ; \\ -F^{(3)}(0) &= \frac{6[(\alpha+\beta)^3 + \alpha\nu(2\alpha+2\beta+\nu)]}{\beta^3\nu^3} . \end{aligned} \tag{7.35}$$

where, $F^{(k)}(0) = F^{(k)}(s)|_{s=0}$.

Model 1

There are several possible ways of fitting the IPP. One way is to equate the computed moments of the inter-departure intervals to those of the IPP, and solve for the unknowns, i.e. the parameters of the IPP. Hence, the following non-linear simultaneous equations in α, β and ν, are to be solved for the latter.

$$\frac{\alpha+\beta}{\beta\nu} = d_1 \; ; \tag{7.36}$$

$$\frac{2[(\alpha+\beta)^2+\alpha\nu]}{\beta^2\nu^2} = d_2 \; ; \tag{7.37}$$

$$\frac{6[(\alpha+\beta)^3+\alpha\nu(2\alpha+2\beta+\nu)]}{\beta^3\nu^3} = d_3 \; . \tag{7.38}$$

The solution for the equations given above is,

$$\begin{aligned}
\alpha &= \frac{9(d_2-2d_1^2)^3}{(6d_1^3-6d_1d_2+d_3)(2d_1d_3-3d_2^2)} \; ; \\
\beta &= \frac{3(d_2-2d_1^2)}{6d_1^3-6d_1d_2+d_3} \; ; \\
\nu &= \frac{2(6d_1^3-6d_1d_2+d_3)}{2d_1d_3-3d_2^2} \; .
\end{aligned} \tag{7.39}$$

Model 2

One other way of fitting the IPP, is to equate only the first two computed moments of the inter-departure interval to those of the IPP, respectively. The other equation is got by equating the average server failure and repair cycle time to the average cycle time of the IPP. Hence, we use equations (7.36) and (7.37). And, the third equation is,

$$1/\alpha + 1/\beta = 1/\xi + 1/\eta \; . \tag{7.40}$$

Then, by solving equations (7.36), (7.37) and (7.40), α, β and ν are computed appropriately, as,

$$\begin{aligned} \nu &= \frac{-b-\sqrt{b^2-4ac}}{2a}; \\ \alpha &= \frac{\eta\nu\xi d_1}{\eta+\xi}; \\ \beta &= \frac{\eta\nu\xi d_1}{(\eta+\xi)(d_1\nu-1)}; \end{aligned} \tag{7.41}$$

where,

$$\begin{aligned} a &= \frac{-2d_1(\eta+\xi)+d_2\eta\xi-2d_1^2\eta\xi}{d_1^2}; \\ b &= \frac{4(\eta+\xi)}{d_1^2}; \\ c &= \frac{-2(\eta+\xi)}{d_1^3}. \end{aligned} \tag{7.42}$$

7.3.3 Validation of the departure model

The approximations, Model 1 and Model 2, of representing the departure process by an IPP, can be very useful. This is because they are computationally simple models. But it is necessary to see how good these models are. Several complementary ways of validation are posible. In this section, we deal with one validation experiment.

Let the departure process, modelled as an IPP by Model 1, be known as IPP^1. Let the one by Model 2 be referred as IPP^2. The validation is done by comparing the 4^{th} moment of the actual inter-departure time, to that of IPP^1, i.e. in Model 1. And, in the case of Model 2, comparing 3^{rd} and 4^{th} moments of the departure process to those of IPP^2, respectively.

We shall consider the same example system as in section 7.2.1. This system has two control parameters, σ and θ.

If the parameters of the IPP are (ν, α, β), then the first 3 moments of the inter-departure interval of the IPP are given by the equations (7.35). The 4^{th} moment of this inter-departure interval of this IPP is given by,

$$F^{(4)}(0) = \frac{24[(\alpha+\beta)^4 + 3\alpha\nu(\alpha+\beta)^2 + \alpha\nu^2(3\alpha+2\beta+\nu)]}{\beta^4\nu^4} \tag{7.43}$$

Hence, the 4th moment of this inter-departure interval of IPP^1 and IPP^2 are obtained by this equation (7.43).

Now, let us find the actual 4th moment of this inter-departure interval. $\Upsilon^{(4)}(0)$ and $\Omega^{(4)}(0)$ can be found by differentiating 4 times successively, the respective equations (7.10, 7.11), with respect to s, and finding their values at $s = 0$. When that is done, we have,

$$\Upsilon^{(4)}(0) = \frac{24[(\eta+\mu+\xi)^4 - 3\eta\mu(\eta+\mu+\xi)^2 + \eta^2\mu^2]}{\eta^4\mu^4} \tag{7.44}$$

$$\Omega^{(4)}(0) = \frac{24[(\eta+\xi)^4 + 3\mu\xi(\eta+\xi)^2 + \mu^2\xi(2\eta+\mu+3\xi)]}{\eta^4\mu^4} \tag{7.45}$$

To get $\Gamma_i^{(4)}(0)$'s, differentiate the equations (7.14, 7.15) 4 times successively and let $s = 0$. Then, we have

$$\Gamma_i^{(4)}(0) = \frac{\sigma_i\Upsilon^{(4)}(0) + \eta\Gamma_{T+i}^{(4)}(0) - 4\Gamma_i^{(3)}(0) + \sum_{k=0\,\&\,k\neq i}^{T-1}\theta_{i,k}\Gamma_i^{(4)}(0)}{\sigma_i+\eta+\theta_i}\ ; \quad i = 0,1,\ldots,T-1 \tag{7.46}$$

$$\Gamma_{T+i}^{(4)}(0) = \frac{\sigma_i\Omega^{(4)}(0) + \xi\Gamma_i^{(4)}(0) - 4\Gamma_{T+i}^{(3)}(0) + \sum_{k=0\,\&\,k\neq i}^{T-1}\theta_{i,k}\Gamma_{T+i}^{(4)}(0)}{\sigma_i+\xi+\theta_i}\ ; \quad i = 0,1,\ldots,T-1 \tag{7.47}$$

Equations (7.46, 7.47) are $2\cdot T$ linear equations and can be solved for the $2\cdot T$ unknowns, $\Gamma_i^{(4)}$'s.

Then, d_4, the actual 4th moment of the inter-departure time, is given by,

$$d_4 = \Delta^{(4)}(0) = \sum_{i=0}^{T-1}\pi_{i,0}\Gamma_{T+i}^{(4)}(0) + (1-\pi_0)\Omega^{(4)}(0)\ . \tag{7.48}$$

$\frac{|(F^{(4)}(0)-d_4)|\cdot 100}{d_4}$ is the percentage error in the 4^{th} moment of the IPP, in both the models. And, $\frac{|(F^{(3)}(0)-d_3)|\cdot 100}{d_3}$ is the percentage error in the of 3^{rd} moment, in case of Model 2.

These percentage errors in the 3^{rd} (in case of Model 2), and 4^{th} moments (in case of Models 1 & 2), of the IPP, are plotted for varying values of σ, keeping $\theta = 1.0$. Figure 7.2 shows these plots. Clearly, the errors in Model 1 are moderately small.

In Figure 7.3, only Model 1 is considered. σ is fixed, and θ is varied over a large range and represented on a logarithmic scale. The errors in the 4^{th} moments are plotted against θ. And, all this is done, for different values of σ. The errors are again reasonably small.

Hence, it may be concluded Model 1 fares well in this validation experiment. However, more and different validation experiments are desirable.

7.4 Conclusions

Bursty arrivals are much more common than Poisson arrivals, in practical systems. The MMPP is an efficient, mathematically tractable and fairly accurate representative model, widely used to represent bursty arrivals. A queue with breakdowns and MMPP arrivals is considered. Firstly, the spectral expansion solution is derived for this system, for steady state probabilities and performance measures. A numerical example and, through that, the necessity of representing the burstiness, are illustrated.

The spectral expansion solution can be extended to the cases of, (*i*) multiserver systems, instead of a single server, (*ii*) general Markovian arrival process (MAP) [40] in place of MMPP, (*iii*) phase-type services, and (*iv*) finite buffer instead of unbounded queueing capacity. Such models have a great deal of applicability in B-ISDN research and design.

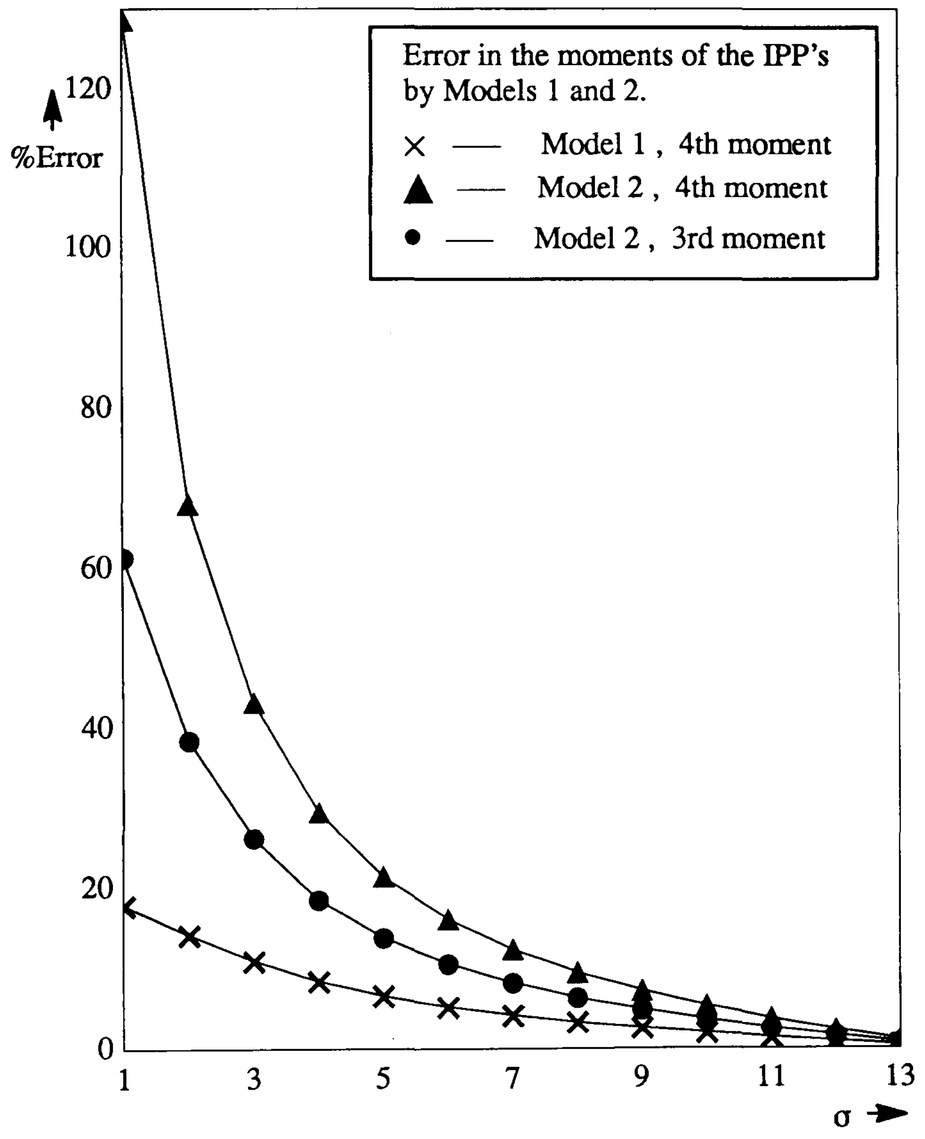

Figure 7.2: Evaluation of the departure IPP: Models 1 & 2.

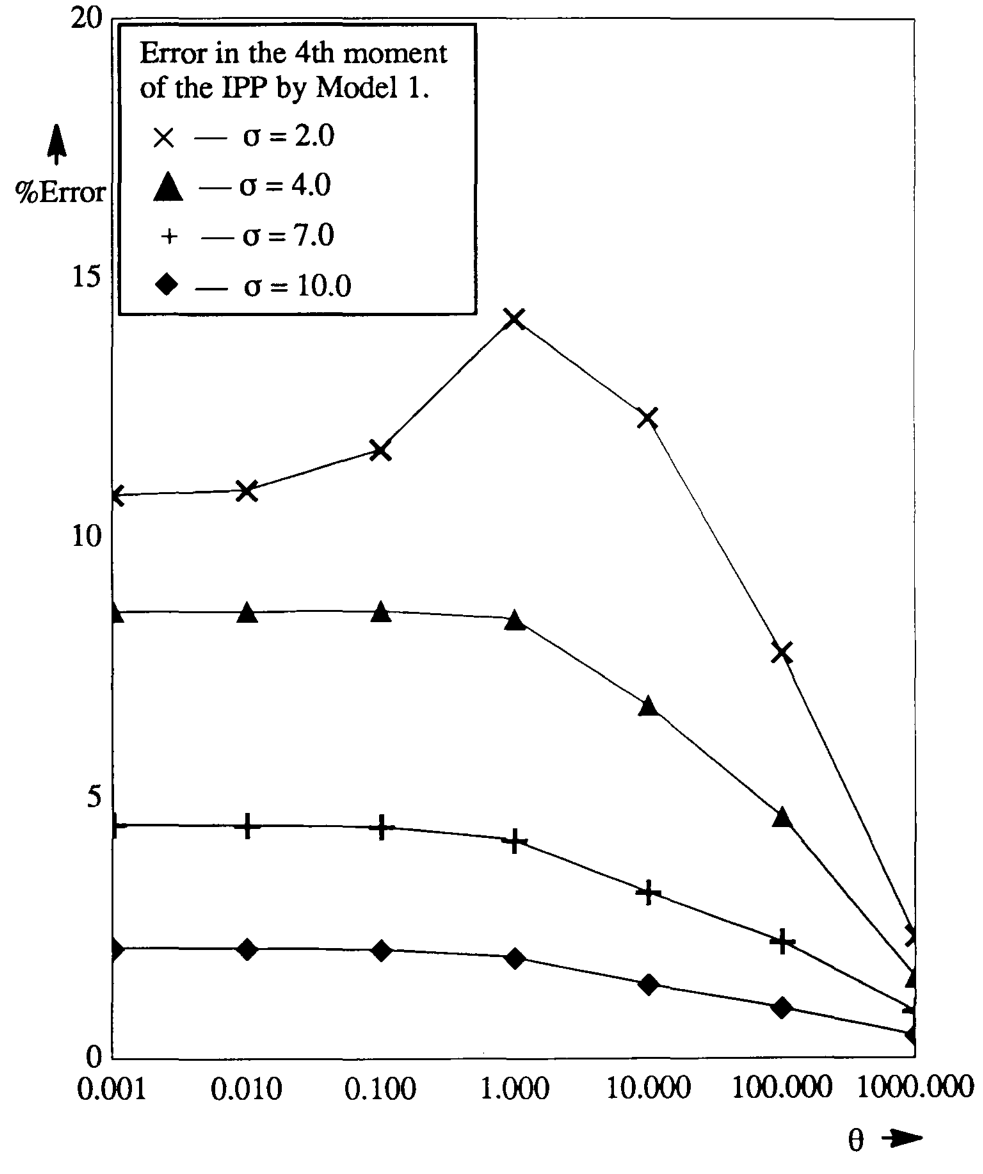

Figure 7.3: Evaluation of the departure IPP: Model 1.

A mathematically simple departure process model is conceived. That is an interrupted Poisson process (IPP), a renewal process. The model parameters are obtained by equating the first 3 moments of the IPP to the corresponding actual moments. A validation procedure for this model, is also given. That model seems to be a fairly good fit. However, further extensive validation is desirable. This may be a good model, even in the case when the server is *reliable*, i.e. without breakdowns.

The departure process modelling can be extended to the cases, (*ii*), (*iii*) and (*iv*), mentioned above. In the case of (*i*), i.e. extension to a homogeneous multiserver system instead of single server, if K is the number of servers, then an MMPP with $K + 1$ phases is a reasonable guess for the departure process model.

7.5 Contributions

This chapter has two contributions. They are,

- A queue with breakdowns and Markov-modulated arrivals, is modelled and analysed, by spectral expansion method, for steady state probabilities and performance measures. A numerical example is presented, as an illustration.

- The departure process is modelled and approximated to an IPP. The parameters of that IPP are derived, from the parameters of the system. This approximation is validated through a procedure. The results of that validation are shown in figures.

These contributions are used in our paper [10].

Chapter 8

Open Queueing Networks with Breakdowns and Repairs

8.1 Introduction

Complex performability problems often require several other systems engineering techniques, along with the spectral expansion method, for wholesome solutions. Two such problems were considered in Chapters 6 and 7. This chapter too deals with one such complex performability problem: modelling and analysis of open queueing networks in which the servers are prone to breakdowns and repairs.

Open queueing networks are models for networks of computers, and modern communication networks consisting of inter-connected nodes. These models are very useful in the analysis of systems, ranging from the telephone networks of today to the information superhighways of tommorrow. Consequently, there has been a great deal of research effort in the performance analysis of open queueing networks, over the last few decades. The pioneering work of Jackson [30] has given rise to simple and efficient solutions for networks, under restricted conditions. Such networks have product form solution [2]. However, many practical networks, e.g. networks with finite capacity, or, with general arrival and service processes, or, with complex priority services, do not fall into the category of product form or BCMP networks. Usually approximate analytical solutions, supported by validation through simulation, are sought or suggested in these cases.

When the servers in a network are prone to breakdowns and repairs, that network does not have a product form solution. The irregularities caused by server breakdowns and repairs do have a great effect on the performance and dependability of the network. Hence, the analysis and solution of such networks is important. So far, there have not been satisfactory solutions to this problem.

In this chapter, we take up this problem of analysing open queueing networks with server breakdowns and repairs, for steady state performance and dependability. We develop several approximate models having different comptational efficiency-accuracy trade offs. We make use of the spectral expansion method, and other systems engineering techniques, wherever necessary, to accomplish this task.

This chapter is organised as follows. The next section defines the problem. Section 8.3 describes a model based on the reduced work rate approximation [54, 64]. In this approximation, each server with breakdowns and repairs is replaced by an equivalent *reliable* server, that is with reduced work rate, to take into account the loss of working time caused by breakdowns. The resulting network is one of product form, and is solved. The results of this model are compared with those obtained by long simulation runs. Necessary conclusions are drawn.

Section 8.4 deals with another model, termed Poisson approximation. In this model, the total flow of jobs to a server, resulting from external arrivals and arrivals from other servers, is approximated or assumed to be Poisson. This is a computationally efficient model. The results, when compared with simulation, seem to be good for large networks. However, for small networks, the results of this model are not quite accurate. Some experimental results are plotted, and analysed.

Two other models are also conceived. These are computationally more involved, but achieving better accuracy, and especially suitable for small networks. Section 8.5 presents one of these, called the MMPP model, involving some approximations and assumptions. In it, the departures from each server are assumed to form, or, approximated by an interrupted Poisson process (IPP). And, the job flows to a server from other servers, and the external arrivals to this server, are all assumed to be independent. With these assumptions, the overall arrivals to a server are shown to form an MMPP process. The network is then modelled as a coupled network of several MMPP/M/1 queues with breakdowns and repairs. This network is solved by an efficient iterative procedure, for performance and dependability measures. This model has given more accurate results compared to the Poisson approximation.

The fourth model is termed as the joint-state model. This is developed in section 8.6. In this, the overall arrival rate to a server is assumed to depend on the joint-state of all the servers. During any given joint-state of the servers, the overall arrival rate to each server is assumed constant. Each queue is, then, a Markov modulated system and the network is a coupled network of Markov modulated queueing systems. This model is solved by an efficient iterative procedure. The results of this model are also found to be more accurate, compared with the Poisson approximation.

8.2 The system

The system to be analysed is an open queueing network of K nodes, numbered as, $1, 2, \ldots, K$. Jobs enter the network, as external arrivals at one or more nodes. Let σ_k, $k = 1, 2, \ldots, K$, be the arrival rate of jobs into node k, from external sources. We assume these external arrivals are Poisson. Each node has a single server that serves jobs one at a time. The servers are prone to failures, and when a server is broken it is set into repair process immediately. Service times, failure times and repair times of the servers are exponentially distributed. Let μ_k, ξ_k and η_k be the service, failure and repair rates respectively, of server k. The queueing capacity at each node is unbounded. When a job's service is interrupted due to the server failure, that service is resumed immediately after that server is repaired. The policy concerning services interrupted by breakdowns is either *resume* from the point of interruption, or, *repeat with re-sampling*. After a job is serviced at node k, it is routed to node l, with a probability $q_{k,l}$, for service at server l. If jobs are never routed from node k to node l, then $q_{k,l} = 0$. It is assumed, without loss of generality, as far as the queue length distributions are concerned, $q_{k,k} = 0$ $(k = 1, 2, \dots, K)$. Also, $q_{k,K+1} = 1 - \sum_{l=1}^{K} q_{k,l}$, is the probability with which a job leaves the system, after its service at node k. We assume, the exit probability, $q_{k,K+1}$, is non-zero for at least one value of k. Q is the routing probability matrix of size $K \times K$, such that, $Q(k,l) = q_{k,l}$ $(1 \le k, l \le K)$. This is the system to be analysed. The system is shown in Figure 8.1. All performance measures to be considered are of steady state, unless specified differently.

Let $\hat{\sigma}_k$ be the total average flow rate of jobs to node k, consisting of external arrivals and arrivals from all other nodes. Then, we can write,

$$\hat{\sigma}_k = \sigma_k + \sum_{l=1}^{K} \hat{\sigma}_l q_{l,k} \; ; \; k = 0, 1, \dots, K \, . \tag{8.1}$$

Let $\boldsymbol{\sigma}$ and $\hat{\boldsymbol{\sigma}}$ be the row vectors, $(\sigma_1, \sigma_2, \dots, \sigma_K)$ and $(\hat{\sigma}_1, \hat{\sigma}_2, \dots, \hat{\sigma}_K)$, repectively. Let E_K be the unit matrix of size $K \times K$. Then, equations (8.1) an also be written as,

$$\hat{\boldsymbol{\sigma}}(Q - E_K) = -\boldsymbol{\sigma} \, . \tag{8.2}$$

Equations (8.1) is a system of K linear equations in K unknowns. $(Q - E_K)$ is non-singular, since $q_{k,K+1}$ is non-zero at least for one value of k. Hence, $\hat{\boldsymbol{\sigma}}$ can be computed by solving these equations.

Let $\hat{\mu}_k$ be the effective average service rate of server k, taking into account the

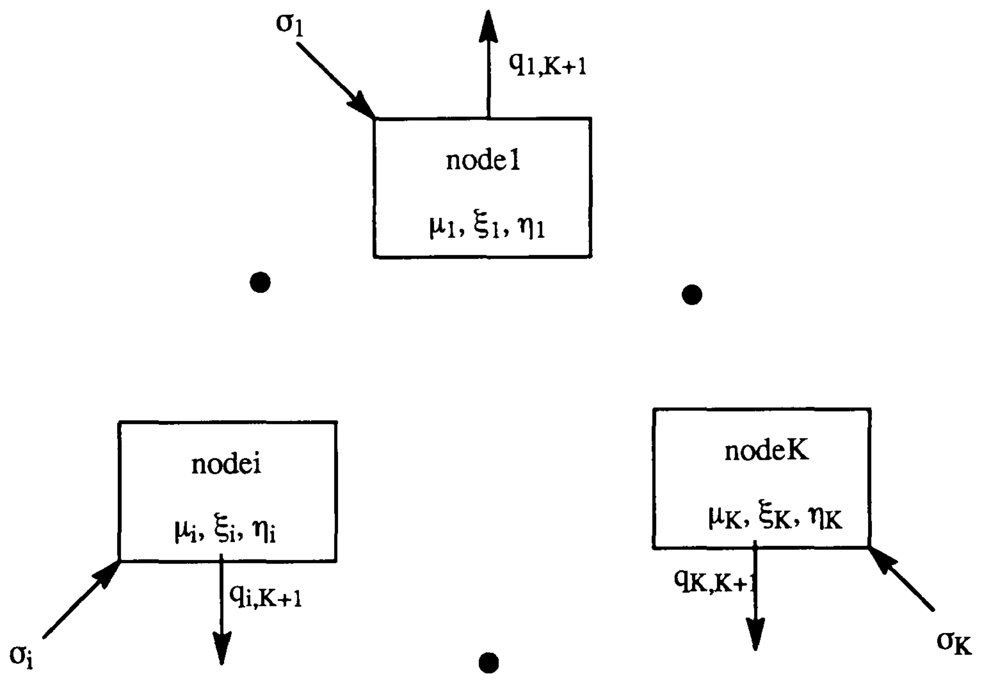

Figure 8.1: Open queueing network with breakdowns and repairs.

loss due to breakdowns and repairs. Then,

$$\begin{aligned}\widehat{\mu}_k &= \mu_k \cdot \frac{\eta_k}{\eta_k + \xi_k} \\ &= \mu_k \cdot \frac{\eta_k/\xi_k}{\eta_k/\xi_k + 1}\ ; \ k = 1, 2, \ldots, K\ . \end{aligned} \tag{8.3}$$

The system is ergodic if, and only if, for every node, the effective average service rate is greater than the total average flow rate into it. That is, $\widehat{\mu}_k > \widehat{\sigma}_k$ ($k = 1, 2, \ldots, K$) (the same remark on page 101 applies here, regarding a rigorous proof of this ergodicity condition).

Hence, we assume that this ergodicity condition holds good for the following steady state performance modelling.

8.3 Reduced work rate approximation

The reduced work rate approximation (RWR) has been used to analyse single server queues with several priority classes of jobs [54]. Also, Vinod and Altiok [64] have successfully used it to model closed queueing networks with server breakdowns and repairs.

We apply this approximation to our problem. According to this, each server k is approximated or replaced by a reliable server with service rate equal to $\widehat{\mu}_k$. The resulting network is an open queueing network of Jackson type. And, that has product form solution. The total flow rate $\widehat{\sigma}_k$ to server k would then be Poisson, in the approximated network. Hence each node can be solved in isolation, as an M/M/1 queue. Thus, the mean queue length at node k, L_k, is given by,

$$L_k = \frac{\widehat{\sigma}_k}{\widehat{\mu}_k - \widehat{\sigma}_k}\ ; \ k = 1, 2, \ldots, K\ . \tag{8.4}$$

The response time of a job, i.e. the average time the job remains in the network, is given by,

$$R = \frac{\sum_{k=1}^{K} L_k}{\sum_{k=1}^{K} \sigma_k}\ . \tag{8.5}$$

8.3.1 Evaluation of the RWR approximation

In order to find out how well the RWR approximation performs, some experiments are done. We take up a tandem network with two servers. The servers have the same service, failure and repair rates. Hence, $\mu_1 = \mu_2 = \mu$, $\xi_1 = \xi_2 = \xi$, and $\eta_1 = \eta_2 = \eta$. We also have chosen $\eta = 10\xi$, in order to keep η/ξ constant. Server 2 has no external arrivals. Hence, $\sigma_2 = 0$. Jobs enter node 1, as Poisson, with arrival rate σ_1. After their service at node 1, jobs enter node 2 for service by Server 2. Hence, $q_{1,2} = 1.0$. Jobs leave the network after their service at node 2. Hence, $q_{2,3} = 1.0$. For such a network, it is possible compute the mean queue length at node 1, L_1, exactly, using the simple formulae in [42]. Hence, what remains is to estimate L_2. This is done using the RWR approximation, and also by long simulation, for some sets of parameter values. The results of the RWR approximation are compared to the simulation results, since the latter are accurate. Typically, we have fixed $\mu = 2.0$, and varied ξ and σ_1 over considerable ranges. The performance characteristics obtained by RWR approximation and by simulation, are shown in Figures 8.2, 8.3 and 8.4.

Figure 8.2 shows the simulation plots, L_2 vs. σ_1, in three example networks. These examples are chosen with three different values of $\xi = (0.001, 0.1, 10.0)$, indeed of different orders of magnitude. However, in all these three examples, η/ξ remains unchanged. Hence, $\widehat{\mu}_1$ and $\widehat{\mu}_2$ also remain unchanged, though ξ changes. Hence, the RWR approximation gives the same result for L_2, for a given value of σ_1, in all these three examples. That common plot, of the RWR approximation, for these three examples is also shown in the figure. Noticeably, the RWR plot coincides only with one of the simulation plots, the one with $\xi = 10.0$. The other two simulation plots show up very differently from the RWR plot. Hence, it can be seen that the RWR approximation is a very inaccurate model.

In the third example, since $\xi(= 10.0)$ and $\eta(= 100.0)$ are very high, by orders

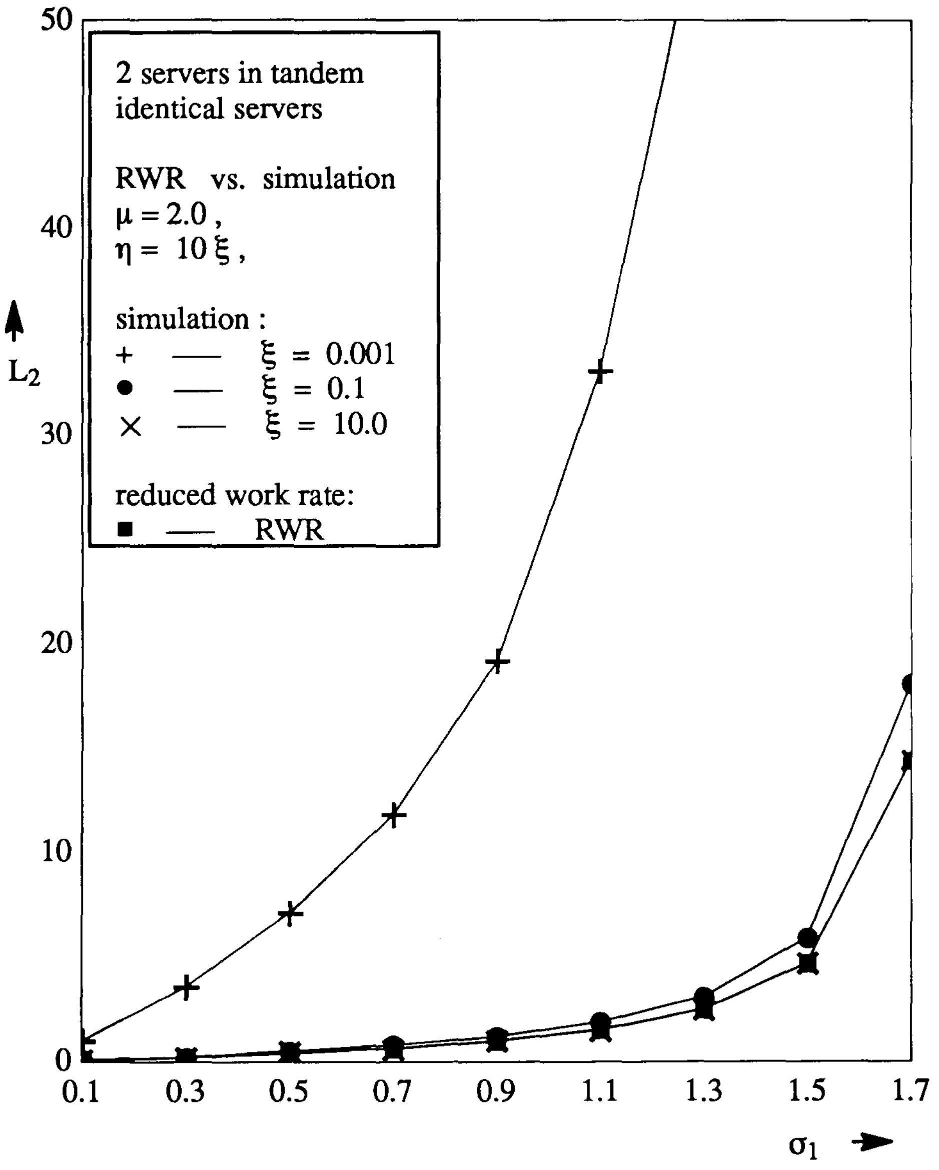

Figure 8.2: Evaluation of the RWR approximation: Experiment 1.

of magnitude, compared to μ and σ_1, no significant build up of queue takes place during the server breakdowns. That is the reason why RWR approximation almost coincides with the simulation result, in this example.

Figures 8.3 and 8.4 show the performance plots for another experiment, Experiment 2. Here, $\mu_1 = \mu_2 = 2.0$, $\xi_1 = \xi_2 = \xi$, and $\eta_1 = \eta_2 = \eta$. Again, the ratio η/ξ is fixed, $\eta = 5\xi$, this time. $\sigma_2 = 0$, and σ_1 is fixed at 1.0. With this, L_2 is plotted against varying ξ. In Figure 8.3, ξ is small, by orders of magnitude, compared to μ and σ_1, and varies linearly on the horizontal axis. Where as, in Figure 8.4, ξ varies exponentially on the horizontal axis, varying from very small values to large values, when compared to the fixed μ and σ_1. In the figures, though plots of other approximations (dealt later) are also given, we are concerned only with the simulation and RWR plots, in this section. It is clear that the RWR approximation differs greatly from simulation results, normally. And, only when ξ and η are much higher than μ and σ_1, the RWR results compare well or coincide with simulation results.

Hence, it can be concluded, for all practical purposes, the RWR approximation is very *inaccurate*.

It is, however, quite surprising and interesting to know the RWR approximation has been reported to perform fairly well in the case of closed queueing networks with server breakdowns and repair [64].

8.4 The Poisson approximation

In this approximation, the total flow of jobs to server k from external arrivals and from all other nodes, is assumed to be Poisson, with rate $\hat{\sigma}_k$. There is some justification of this assumption, in certain cases. There have been results to show that the superposition of many independent and relatively sparse processes converges to Poisson, as the number of component processes tends to infinity [11]. This approx-

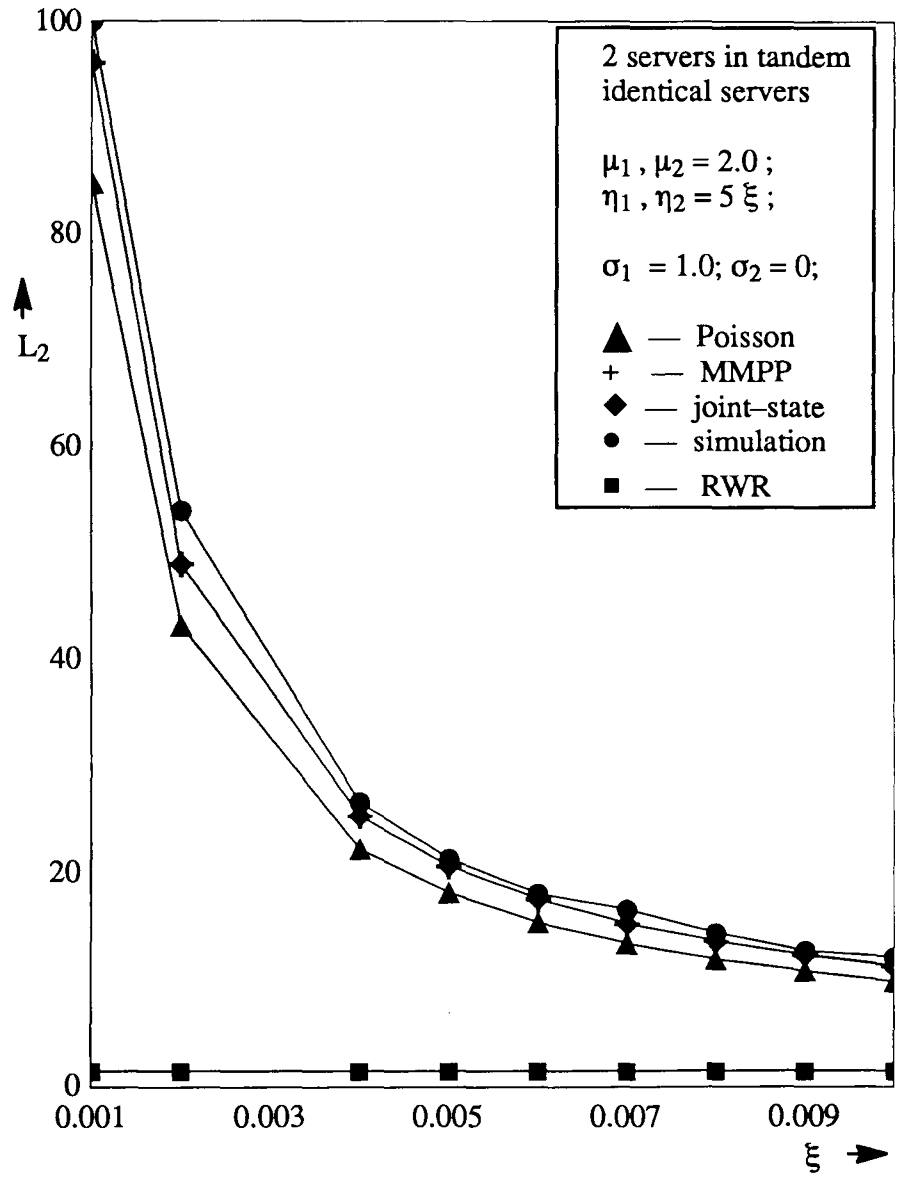

Figure 8.3: L_2 vs. ξ in the two-server network.

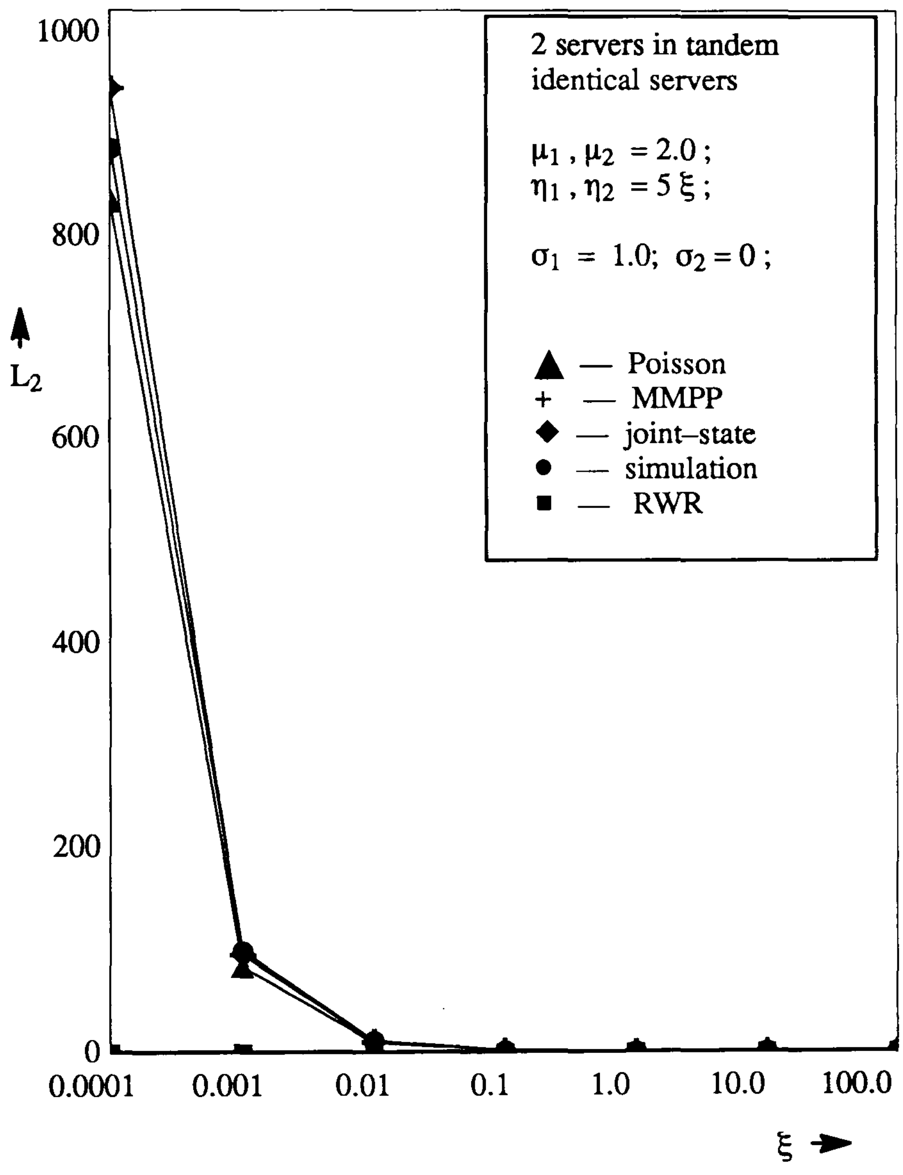

Figure 8.4: L_2 vs. ξ in the two-server network.

imation was reported, but not evaluated in [43].

With the above assumption, the problem reduces to, (i) computing the total average flow rates, $\hat{\sigma}_k$'s, to nodes (ii) Analysing each node separately, as a M/M/1 queue with breakdowns and repairs of the server, for queue length distribution.

For task (ii) above, Mitrani and Avi-Itzhak have derived a simple mathematical formula [42]. Using that formula, L_k's can be computed as,

$$L_k = \frac{\hat{\sigma}_k[(\xi_k + \eta_k)^2 + \mu_k \xi_k]}{(\xi_k + \eta_k)[\eta_k(\mu_k - \hat{\sigma}_k) - \hat{\sigma}_k \xi_k]} \; ; \; k = 1, 2, \ldots, K \; . \tag{8.6}$$

Once, L_k's are known, the equation (8.5) can be used to compute the response time.

8.4.1 Evaluation of the Poisson approximation

Evaluation study is performed on a number of randomly generated sample or example networks. In the first experiment, i.e. Experiment 1, random networks of four different sizes, $K = (3, 5, 10, 15)$, are generated. For each of those sizes, eight different sample or example networks are generated.

Each sample network is generated, by choosing its parameter values randomly, as follows: First, ξ_k's are generated by sampling uniformly as, $\xi_k \in (0.001, 0.01)$. Then, η_k's are generated as, $\frac{\eta_k}{\xi_k} \in (1.0, 10.0)$. The external arrivals are generated as, $\sigma_k \in (0.1, 3.0)$. The exit probabilities, $q_{k,K+1}$'s, are generated as, $q_{k,K+1} \in (0.1, 0.5)$. Once the exist probability of server k is chosen, the transition probabilities from server k to all other servers are determined randomly. This is done by splitting the quantity, $1 - q_{k,K+1}$, unformly and randomly, into $K - 1$ parts. These, latter $K - 1$ quantities form the $K - 1$ transition probabilities, $q_{k,l}$ $(1 \leq l \leq K, l \neq k)$. With these parameter values, the total average flow rates to servers, $\hat{\sigma}_k$'s, are determined, solving the equations (8.1). The service rates of the servers, μ_k's are now chosen as, $\frac{\mu_k \eta_k}{\hat{\sigma}_k(\xi_k + \eta_k)} \in (1.1, 3.0)$. The sample network generation is now complete.

Each sample network is analysed using the Poisson approximation and also by a long simulation run, the latter producing accurate results. These results are, then, compared. Let $R_{poisson}$ and $R_{simulation}$ be the response times computed by the Poisson approximation and simulation respectively. Then $\frac{|(R_{poisson}-R_{simulation})|\cdot 100}{R_{simulation}}$ is the percentage error in the response time computation, using the Poisson approximation.

Figure 8.5 shows the percentage error in response time computation by the Poisson approximation, vs. the size of the network, K. Each point (dot) concerns with one sample network. There are eight sample networks for each $K = (3, 5, 10, 15)$. It is clearly evident, the larger the size of the network is, the Poisson approximation seems to fit better. The line joining the average of these errors for each K, is also shown.

To confirm this trend, another experiment, Experiment 2, also is conducted. This is done in a slightly different way, and on just one sample network of size, $K = 20$. The network parameter values are generated again randomly as follows: First ξ_k's are chosen as, $\xi_k \in (0.0001, 0.001)$; then, η_k's as, $\frac{\eta_k}{\xi_k} \in (10, 10.0)$; the exit probabilities as, $q_{k,K+1} \in (0.1, 0.5)$. The transition probabilities from each server k to the other servers are then sampled, just as in the case of Experiment 1 of this section. This is done by splitting the quantity, $1 - q_{k,K+1}$, randomly and uniformly, into $q_{k,l}$'s $(1 \leq l \leq K, l \neq k)$.

The external arrivals are a single Poisson arrival stream of jobs, with arrival rate σ. σ is the variable in this experiment. This stream splits into K streams, σ_k $(k = 1, 2, \ldots, K)$, unformly and randomly. This is done, by randomly choosing the split probabilities, δ_k $(k = 1, 2, \ldots, K)$, of σ, uniformly. Also, $\sum_{k=1}^{K} \delta_k$ should be equal to 1.0. These chosen or sampled split probabilities are now fixed. Stream k, with arrival rate $\sigma_k = \sigma\delta_k$, enters node k, as an external Poisson arrival stream. The maximum value of σ is fixed as 10.0.

The service rates, μ_k's, are chosen as follows. With $\sigma = 10.0$, the maximum value

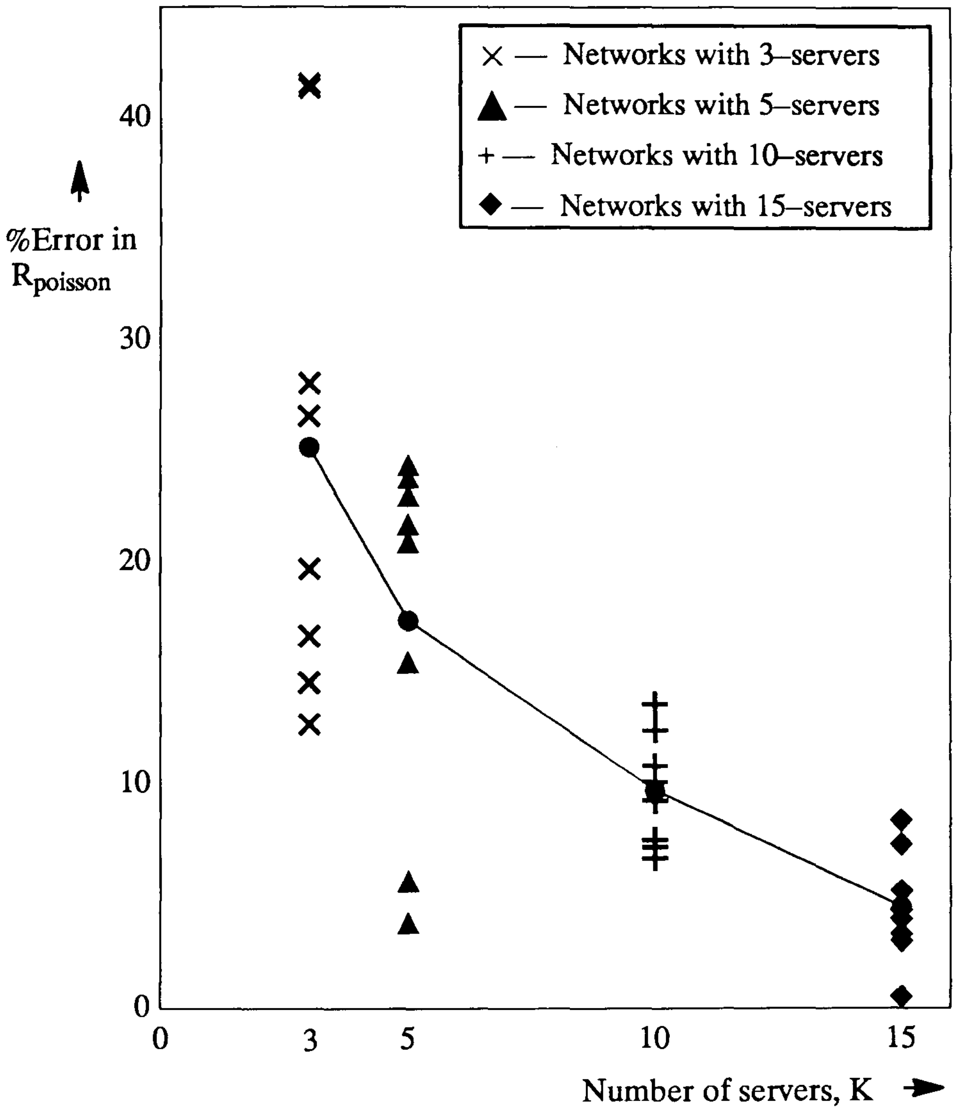

Figure 8.5: Evaluation of the Poisson approximation: Experiment 1.

of σ, the streams to individual nodes, σ_k's, are computed. Then, using equations (8.1), the total average flows to the nodes, $\widehat{\sigma}_k$'s, are computed. Then, the service rates, μ_k's, of servers are randomly chosen as, $\frac{\mu_k \eta_k}{\sigma_k(\xi_k+\eta_k)} \in (1.01, 1.5)$. With this, all the parameters of the network are fixed. The network would be ergodic, if $\sigma < 10.0$.

This network is analysed by the Poisson approximation, and also, by long simulation, for varying values of σ. All the other network parameters are kept fixed. The response times $R_{poisson}$ and $R_{simulation}$ are computed, in each case.

Figure 8.6 shows the response time plots against σ, the total external arrival rate, for the Poisson approximation and for the simulation. The confidence intervals of the simulation results are also shown. The Poisson approximation seems to be a good fit, in this experiment. The error in the response time computation, for this example system, is even smaller than the average error for the example systems of size $K = 15$ considered in Experiment 1.

It can now be said that the Poisson approximation gives fairly accurate results in large networks in which the nodes receive arrival streams from a number of other nodes. That is because, the superposition of all those arrivals to a node, including the external arrivals, can then be approximated as Poisson for the sake of queue length computations, without losing a great deal of accuracy.

Also in networks, small or large, if the external Poisson arrivals to each node are much larger compared to the sum of other arrivals (from other nodes) to that node, then also the Poisson approximation would work well. This situation occurs if the exit probabilities of nodes are large, compared to the transition probabilities to the other nodes.

Two more approximate models are presented in the following sections. In these models, the attention is towards small networks, since Poisson approximation has given satisfactory results for large networks.

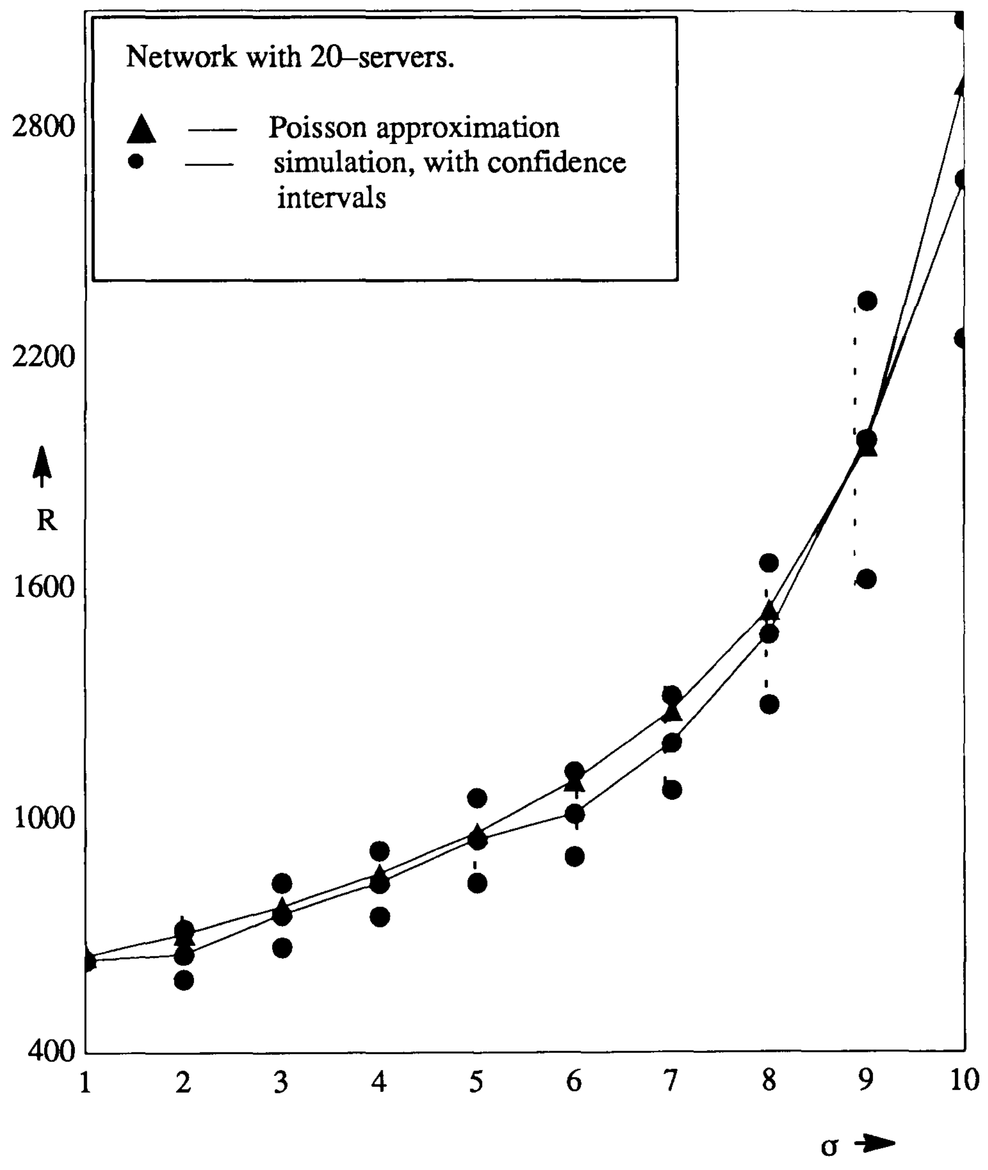

Figure 8.6: Evaluation of the Poisson approximation: Experiment 2.

8.5 The MMPP model

The Poisson approximation does not seem to work well for networks, with considerably small number of nodes. This is because, in small networks, the nodes receive fewer number of arrival streams, external as well as from other nodes. The superposition of these streams does not tend to be Poisson. The MMPP approximation or the MMPP model, is developed in order to improve over the Poisson approximation, by accounting for this trend. Let S_k be the set of nodes to which the departures from node k are routed. That is, if $q_{k,l} > 0$ then, node $l \in S_k$. Let G_k be the set of nodes from which node k receives jobs. That is, if $q_{l,k} > 0$ then, node $l \in G_k$. Let s_k be the cardinality of S_k and, g_k that of G_k.

The MMPP model is based on the following underlying assumptions and approximations:

(a) The departure process from node k ($k = 1, 2, \ldots, K$), is approximated by an interrupted Poisson process(IPP), with parameters $(\nu_k, \alpha_k, \beta_k)$. The definition of these parameters is the same as in Chapter 7. We name this process, individually as, IPP_k.

(b) The split process, of the departure IPP from node k, entering node l (node $l \in S_k$), is also assumed or approximated to an IPP with parameters $(\nu_{k,l}, \alpha_{k,l}, \beta_{k,l})$. We name this process, individually as, $IPP_{k,l}$.

(c) The departure IPP's from all the nodes, the arrival IPP processes to all the nodes, and the external Poisson arrival processes to all nodes, are all assumed to be independent of one another.

With these assumptions, node k receives g_k independent IPP's from the nodes belonging to G_k, plus, an independent external Poisson arrival stream. The superposition of these independent streams entering node k, is then an MMPP. We name this process as, $MMPP_k$. Each node, thus, reduces to an MMPP/M/1 queue with with server breakdowns and repairs. That can be solved using the model and analysis described in Chapter 7. It can be noted, if $g_k = 0$, then $MMPP_k$ can be considered as merely the Poisson process with rate σ_k.

Now, what need to be determined or estimated are, (i) parameters of IPP_k, the departure process from node k, $(\nu_k, \alpha_k, \beta_k)$ $(i = 1, 2, \ldots, K)$, (ii) parameters of $IPP_{k,l}$, the arrival process at node l from node k, $(\nu_{k,l}, \alpha_{k,l}, \beta_{k,l})$ $(1 \leq k, l \leq K\ ;\ q_{k,l} > 0)$, and, (iii) parameters of $MMPP_k$'s $(k = 1, 2, \ldots, K)$.

This is done as follows.

8.5.1 Probablisic splitting of IPP_k

The parameters of IPP_k, the departure IPP from node k, are $(\nu_k, \alpha_k, \beta_k)$. This process is split into $s_k (\leq K - 1)$ split processes, each of which enters another appropriate node. The arrival process to node $l\,(\in S_k)$, from node k, approximated as an IPP, is obtained by sampling the departures from node k with probability $q_{k,l}$. Let $\Delta_k(s)$ be the Laplace transform of the distribution of the inter-departure interval of IPP_k. And, let $\Delta_{k,l}(s)$ be the Laplace transform of the distribution of the inter-arrival time of $IPP_{k,l}$. Then, we have,

$$\Delta_k(s) = \frac{\nu_k s + \nu_k \beta_k}{s^2 + s(\nu_k + \alpha_k + \beta_k) + \nu_k \beta_k}\ ;\ k = 1, 2, \ldots, K\,. \tag{8.7}$$

$\Delta_{k,l}(s)$ can be expressed in terms of $\Delta_k(s)$ and $q_{k,l}$ as follows. Let, at time t_1, there be an arrival at node l from node k. Let the next or subsequent arrival at node l, from node k, be at time t_2. During the interval between these two events, there may be 0 or more departures (not counting the ones at t_1 and t_2) from node k that were not routed to node l, as a result of probablistic sampling. The probability that such departures from node k are exactly n is given by $(1 - q_{k,l})^n q_{k,l}$. In such a case, the Laplace transform of the distribution of the random variable $(t_2 - t_1)$ is

$\Delta_k^{n+1}(s)$. Therefore, the Laplace transform of the distribution of the time $(t_2 - t_1)$, for any value of n, can be written as,

$$\begin{aligned}\Delta_{k,l}(s) &= q_{k,l}\Delta_k(s) + (1-q_{k,l})q_{k,l}\Delta_k^2(s) + (1-q_{k,l})^2 q_{k,l}\Delta_k^3(s) + \ldots \\ &= \sum_{n=0}^{\infty}(1-q_{k,l})^n q_{k,l}\Delta_k^{n+1}(s)\,. \end{aligned} \tag{8.8}$$

From equations (8.7, 8.8), $\Delta_{k,l}(s)$ can be derived as,

$$\Delta_{k,l}(s) = \frac{q_{k,l}\nu_k s + q_{k,l}\nu_k\beta_k}{s^2 + (q_{k,l}\nu_k + \alpha_k + \beta_k)s + q_{k,l}\nu_k\beta_k}\,. \tag{8.9}$$

We have assumed the arrival process to node l from node k is an IPP. Hence, $\Delta_{k,l}(s)$, derived as above, is the Laplace transform of the inter-arrival interval of $IPP_{k,l}$. Looking at its structure, it can be inferred the parameters of this $IPP_{k,l}$, $(\nu_{k,l}, \alpha_{k,l}, \beta_{k,l})$, can be given by,

$$\nu_{k,l} = q_{k,l}\nu_k\ ,\ \alpha_{k,l} = \alpha_k\ ,\ \beta_{k,l} = \beta_k\ ;\ 1 \le k,l \le K\,;\ k \ne l\,. \tag{8.10}$$

8.5.2 Constructing $MMPP_k$

The arrivals at node k are the arrivals from all the nodes in G_k, that is, $IPP_{l,k}$'s $(q_{l,k} > 0)$, plus, the external Poisson arrivals with rate σ_k. These are $g_k + 1$ independent arrival processes at node k. The superposition of all these arrivals is an MMPP process with $T_k = 2^{g_k}$ phases. This process, individually, is referred as $MMPP_k$. The parameters of $MMPP_k$ can be derived quite easily, using the results in [16].

If $g_k = 0$, then node k has only external arrivals. And, the $MMPP_k$, in this case, can be thought of as the Poisson process with rate σ_k.

If $g_k > 0$ then, let the nodes in G_k be designated as, $(1,k), (2,k), \ldots, (g_k, k)$, in any order. Each of these ordered pairs, (m,k) for any $m\,(\le g_k)$, represents or corresponds to a unique or distinct node $l\,(\in G_k)$, and hence, is synonymous with the latter. Also in such a case, i.e. if node l and node (m,k) are synonymous, let

$\nu_{(m,k)} = \nu_{l,k}$, $\alpha_{(m,k)} = \alpha_{l,k}$ and $\beta_{(m,k)} = \beta_{l,k}$. Similarly, let $IPP_{(m,k)}$ mean the same as $IPP_{l,k}$. Let $\Sigma_{(m,k)}$, $\Theta_{(m,k)}$ be matrices of size 2×2 each, given by,

$$\Sigma_{(m,k)} = \begin{bmatrix} \nu_{(m,k)} & 0 \\ 0 & 0 \end{bmatrix} \; ; \; m = 1, 2, \ldots, g_k \; ; \tag{8.11}$$

$$\Theta_{(m,k)} = \begin{bmatrix} -\alpha_{(m,k)} & \alpha_{(m,k)} \\ \beta_{(m,k)} & -\beta_{(m,k)} \end{bmatrix} \; ; \; m = 1, 2, \ldots, g_k \; . \tag{8.12}$$

Let the phases of the $MMPP_k$ be numbered as, $0, 1, \ldots, T_k - 1$, with T_k phases. Let the arrival rate in phase n $(n = 0, 1, \ldots, T_k - 1)$ be $\sigma_{k,n}$. Let Σ_k be the diagonal matrix, of size $T_k \times T_k$, whose n^{th} diagonal element is $\sigma_{k,n}$. Let Θ_k be the generator matrix, of size $T_k \times T_k$, of $MMPP_k$. Now, we need to relate Σ_k and Θ_k to the parameters of the $IPP_{l,k}$'s (node $l \in G_k$). From the results in [16], that can be written as,

$$\Sigma_k = \Sigma_{(1,k)} \oplus \Sigma_{(2,k)} \oplus \ldots \oplus \Sigma_{(g_k,k)} \; + \; \sigma_k E_{T_k} \tag{8.13}$$

$$\Theta_k = \Theta_{(1,k)} \oplus \Theta_{(2,k)} \oplus \ldots \oplus \Theta_{(g_k,k)} \; ; \tag{8.14}$$

where, $\oplus$ stands for the Kronecker sum [16], and E_{T_k} is the unit matrix of size $T_k \times T_k$.

Thus, the parameters of $MMPP_k$ are now expressed in terms of the parameters of all the IPP_l's (node $l \in G_k$).

8.5.3 The iterative procedure

So far, we have been able to express the parameters of all the $MMPP_k$'s and those of the $IPP_{k,l}$'s, in terms of the parameters of all the IPP_k's $(k = 1, 2, \ldots, K)$. If the parameters of the $MMPP_k$'s are computed, then the steady state mean queue-lengths, utilizations and many other dependability measures can be computed, using the analysis in Chapter 7. Hence, we need to compute, $(\nu_k, \alpha_k, \beta_k)$ $(k = 1, 2, \ldots, K)$.

We have used an iterative procedure to compute these parameters and the performance measures. That procedure is as follows:

(1) Compute the total average arrival rates to all the nodes, $\widehat{\sigma}_k$'s ($k = 1, 2, \ldots, K$), using the equations (8.1). Solve each node k as an MMPP/M/1 queue with breakdowns and repairs. In this first iteration, this is done considering the $MMPP_k$ is just a Poisson process with rate $\widehat{\sigma}_k$. Compute the queuelengths L_k, for each node k, in this iteration. Also, compute the departure process, IPP_k, from each node k. Go to step (2).

(2) Using the IPP_k's ($k = 1, 2, \ldots, K$) computed in the previous iteration, reconstruct the $MMPP_k$'s in this iteration, using the procedure in sections 8.5.1 and 8.5.2. Again, solve each node k, as an MMPP/M/1 queue with breakdowns and repairs, but with the newly computed or constructed $MMPP_k$ as the arrival process. That is done using the results of Chapter 7. Also, compute the new values of L_k's, in this iteration. Find out, ϵ, the sum of the absolute values, of the differences between the corresponding queuelengths, in this and the previous iteration. That is, $\epsilon = \sum_{k=1}^{K} \mid L_k^{(new)} - L_k^{(old)} \mid$. Go to step (3).

(3) If ϵ, computed in the previous step, is greater than a specified value $\epsilon_{(tol)}$, then, go to step (2) for futher iteration. However, if ϵ is less than $\epsilon_{(tol)}$, then, go to (4).

(4) Stop further iteration. Compute all the required steady state performance and dependability measures using the most recently computed parameters.

We have no proof for the convergence of this iterative procedure, for steady state performance measures. Yet, in all the example networks we have analysed, we have observed fast convergence even for very low values of the specified tolerance, $\epsilon_{(tol)}$.

The evaluation of the MMPP model is presented in section 8.7, along with that of the Joint-state model described below.

8.6 The Joint-state model

Let $O_k(t)$ be the state of the server k at time t. $O_k(t)$ is a binary random variable with values 0, signifying *broken*; or 1, implying *operative*. The joint-state of all the servers is then an ordered K-bit binary random variable, $(O_1, O_2, \ldots, O_K)$. The k^{th} bit from the left is O_k that indicates the state of server k. The total number of these joint-states is 2^K. Let these states be numbered in the decimal system as, $0, 1, \ldots, N$, where $N = 2^K - 1$. The joint-state i, in decimal form, refers to the joint-state that is K-bit binary expansion of i. In it, k^{th} bit from left indicates the state of server k.

The joint-states of the servers, are $N+1$, in total. Let A be the transition rate matrix of these joint-states. A is of size $(N+1) \times (N+1)$. $A(i,l)$ is the transition rate from the joint-state i to the joint-state l $(0 \leq i, l \leq N\,;\, l \neq i)$.

A transition in a joint-state i occurs when an operative server breaks down, or a broken server becomes operative after a repair. Hence, the number of different transitions that are possible from any joint-state i, are K. And, each of these transition rates is one of ξ_k's or η_k's $(k = 1, 2, \ldots, K)$.

For example, for a 3-node network, that matrix A would be as below:

$$\begin{array}{c} (0,0,0) \\ (0,0,1) \\ (0,1,0) \\ (0,1,1) \\ (1,0,0) \\ (1,0,1) \\ (1,1,0) \\ (1,1,1) \end{array} \quad A = \begin{bmatrix} 0 & \eta_3 & \eta_2 & 0 & \eta_1 & 0 & 0 & 0 \\ \xi_3 & 0 & 0 & \eta_2 & 0 & \eta_1 & 0 & 0 \\ \xi_2 & 0 & 0 & \eta_3 & 0 & 0 & \eta_1 & 0 \\ 0 & \xi_2 & \xi_3 & 0 & 0 & 0 & 0 & \eta_1 \\ \xi_1 & 0 & 0 & 0 & 0 & \eta_3 & \eta_2 & 0 \\ 0 & \xi_1 & 0 & 0 & \xi_3 & 0 & 0 & \eta_2 \\ 0 & 0 & \xi_1 & 0 & \xi_2 & 0 & 0 & \eta_3 \\ 0 & 0 & 0 & \xi_1 & 0 & \xi_2 & \xi_3 & 0 \end{bmatrix} . \tag{8.15}$$

Let the joint-state i be expressed in K-bit binary form as, $(o_{1,i}, o_{2,i}, \ldots, o_{K,i})$.

Each $o_{k,i}$ is either 0 or 1. The total transition rate, r_i, from joint-state i is the sum of all the transition rates in the corresponding (i^{th}) row of A. Hence, the average time the set of nodes remain in the joint-state i is $1/r_i$. Also, let the service rate of server k, during the joint-state i, be $\mu_{k,i}$. Then, obviously, $\mu_{k,i} = \mu_k$ if $o_{k,i} = 1$, and, $\mu_{k,i} = 0$ if $o_{k,i} = 0$.

The Joint-state model is based on the following assumptions or approximations.

(a) The total job arrival process to any node k ($k = 1, 2, \ldots, K$), during any joint-state i ($i = 0, 1, \ldots, N$) is Poisson, with rate $\widehat{\sigma}_{k,i}$.

(b) The total job departure process from node k ($k = 1, 2, \ldots, K$), during joint-state i ($i = 0, 1, \ldots, N$) is Poisson, with rate $\widehat{\nu}_{k,i}$.

With the first approximation, each node k can be considered, in isolation, as an MMPP/M/1 queue, with ($N+1$)-phase MMPP arrivals, and service rates dependent on the phase of the arrival process. We call this an MMPP/M/1 correlated queue, or, a Markov-modulated queue. That queue can be solved using spectral expansion procedure for required steady state probabilities and performance measures. That solution is described in section 8.6.1.

The network is, thus, a coupled system of K Markov-modulated queues. The remaining task is to estimate $\widehat{\sigma}_{k,i}$'s and $\widehat{\nu}_{k,i}$'s, appropriately. This is done iteratively in section 8.6.2.

8.6.1 Analysis of node k

As said earlier, each node k is a Markov-modulated queue or an MMPP/M/1 correlated queue. The job arrival process has $N + 1$ phases, and they are synonymous with the joint-states of the servers. The service rate also depends on the phase of the arrival process. In phase i, the arrivals are Poisson with rate $\widehat{\sigma}_{k,i}$ and the service rate is $\mu_{k,i}$. If node k is operative, in the joint-state i, then $\mu_{k,i} = \mu_k$, otherwise $\mu_{k,i} = 0$.

This Markov-modulated queue, of node k, can be represented by the Markov process X of Chapter 2. The parameters of this process X, then, are given as follows: The bounded random variable, $I(t)$, represents the phase of the MMPP, $0, 1, \ldots, N$. The unbounded random variable, $J(t)$ indicates the number of jobs at node k, including the one being served, if any.

The purely lateral transition rate matrices A and A_j's of X are the same as the joint-state transition rate matrix A. The one-step upward transition rate matrices B and B_j's are given by,

$$B = B_j \ (j = 0, 1, \ldots) = diag[\widehat{\sigma}_{k,0}, \widehat{\sigma}_{k,1}, \ldots, \widehat{\sigma}_{k,N}] \ . \tag{8.16}$$

The one-step downward transition rate matrices C and C_j's are given by,

$$\begin{aligned} C &= C_j \ (j = 1, 2, \ldots) = diag[\mu_{k,0}, \mu_{k,1}, \ldots, \mu_{k,N}] \ ; \\ C_0 &= 0 \ . \end{aligned} \tag{8.17}$$

The threshold M is clearly equal to 1. Now, the parameters of X are clearly defined. Hence, the process X and the Markov-modulated queue of node k, can be solved, using the spectral expansion method described in Chapter 2, if the parameters of process X are known. So far, the matrix B is unknown.

Let $p_{i,j}$ be the steady state probability that, at node k, the number of jobs is j and the arrival phase (joint-state of servers) is i. Then, $\widehat{\nu}_{k,i}$'s can be expressed as,

$$\widehat{\nu}_{k,i} = \mu_{k,i} \cdot \left(1 - \frac{p_{i,0}}{\sum_{j=0}^{\infty} p_{i,j}}\right) \ ; \ i = 0, 1, \ldots, N \ . \tag{8.18}$$

Also, $\widehat{\sigma}_{k,i}$'s can be expressed in terms of $\widehat{\nu}_{k,i}$'s as,

$$\widehat{\sigma}_{k,i} = \sum_{l=1}^{K} \widehat{\nu}_{l,i} q_{l,k} + \sigma_k \ ; \ i = 0, 1, \ldots, N \ . \tag{8.19}$$

Thus, what now remains is to estimate the unknown parameters, $\widehat{\sigma}_{k,i}$'s or $\widehat{\nu}_{k,i}$'s. This is done iteratively, and described in the next section.

Notice that, the process X and its parameters, defined in this section, pertain to node k, though we have deliberately avoided the suffix k in these parameters. This is done only for making the notation simple and easily understandable. However, the parameters of the Markov process pertaining to any other node l ($l \neq k$) can be quite different from those of node k.

8.6.2 Iterative solution

The unknowns, $\widehat{\sigma}_{k,i}$'s and $\widehat{\nu}_{k,i}$'s, can be obtained iteratively. We start, i.e in the first iteration, with $\widehat{\sigma}_{k,i} = \widehat{\sigma}_k$. The procedure is as follows:

(1) Compute the total average arrival rates to all nodes, $\widehat{\sigma}_k$'s ($k = 1, 2, \ldots, K$). For each node k, let $\widehat{\sigma}_{k,i} = \widehat{\sigma}_k$ ($i = 0, 1 \ldots, N$). Solve each node k as a Markov-modulated queue, by spectral expansion, as outlined in section 8.6.1. Compute L_k's and the necessary steady state probabilities. Compute, for each node k, $\widehat{\nu}_{k,i}$'s from equation (8.18). Go to step (2).

(2) For each node k, using the $\widehat{\nu}_{k,i}$'s computed in the previous iteration, compute $\widehat{\sigma}_{k,i}$'s, from equation (8.19). Solve again each node k, as a Markov-modulated queue, by spectral expansion, but with the newly computed $\widehat{\sigma}_{k,i}$'s. Compute L_k's and the necessary probabilities. Compute again, for each node k, $\widehat{\nu}_{k,i}$'s from equation (8.18). Compute ϵ as, $\epsilon = \sum_{k=1}^{K} \mid L_k^{(new)} - L_k^{(old)} \mid$. Go to step (3).

(3) If ϵ, computed in the previous step, is greater than a specified value $\epsilon_{(tol)}$, then, go to step (2) for futher iteration. However, if ϵ is less than $\epsilon_{(tol)}$, then, go to (4).

(4) Stop further iteration. Compute all the required steady state performance and dependability measures using the most recently computed parameters.

Just as in the case of MMPP model, we do not have any proof for convergence of this iterative procedure. However, in all our example networks, we noticed fast convergence even for very low values of the specified tolerance, $\epsilon_{(tol)}$.

8.7 Evaluation of the MMPP and the Joint-state models

We have evaluated these models, using long simulation runs. Since the number of parameters of a network is large, even for moderately small values of K, experimental validation of these models is possible only on a limited scale. Yet, we have evaluated dozens of randomly chosen example networks, of sizes $K = 2, 3, 4, 5$. This is done for different parameter ranges. The accuracies observed are quite varied. It is normally observed that the accuracies are relatively lower, if the number of external arrival processes are smaller. Hence, the most challenging ones (rather the worst cases) are, in which there are external arrivals to only one of the nodes. Hence, we have chosen to present here some of those examples in which external arrivals exist only for one of the nodes. Without loss of generality, we number that node as, node 1. In the examples given in this chapter, we have included some our least accurate results, as well.

In the two-server network of Figure 8.3, the MMPP model and the Joint-state model give the same results. Indeed, they perform more accurately, compared to the Poisson approximation.

For $K = 3$, we have considered 3 examples. Figures 8.7 and 8.8 illustrate a 3-server tandem network, i.e. Example 1. The parameters of the network are displayed in the figures. In all the examples, we have taken, $\epsilon_{(tol)} = 0.01$. As said earlier,

$\sigma_2 = 0$ and $\sigma_3 = 0$. In Figure 8.7, the response time, R, is plotted against varying σ_1. The MMPP and the Joint-state models perform extremely well, and better than the Poisson approximation. Figure 8.8 shows L_3 vs. σ_1. We have chosen L_3 in the figure because of all the 3 queuelengths, L_3 is computed in the models, with least accuracy. The MMPP model does best, and the Joint-state model is highly satisfactory. More performance measures (P.M.'s) are displayed in Table 8.7.

Table 8.1: 3-node network: Example 1.

$\sigma_1 = 0.5$				
P.M.s	Poisson	MMPP	Joint-state	Simulation
$L_1 =$	98.2	98.2	98.2	95.0
$L_2 =$	77.6	80.7	80.7	79.7
$L_3 =$	42.5	46.4	45.9	49.5
$R =$	436.5	450.7	449.6	448.4
$\sigma_1 = 1.7$				
P.M.	Poisson	MMPP	Joint-state	Simulation
$L_1 =$	579.3	579.3	579.3	556.9
$L_2 =$	430.4	523.3	523.2	564.8
$L_3 =$	209.2	305.2	288.7	328.9
$R =$	716.9	828.0	818.3	853.3
$\sigma_1 = 2.9$				
P.M.	Poisson	MMPP	Joint-state	Simulation
$L_1 =$	3724.6	3724.6	3724.6	3538.7
$L_2 =$	1992.7	3365.6	3365.5	2883.1
$L_3 =$	647.1	1280.6	1133.2	1256.8
$R =$	2194.6	2886.5	2835.6	2647.8

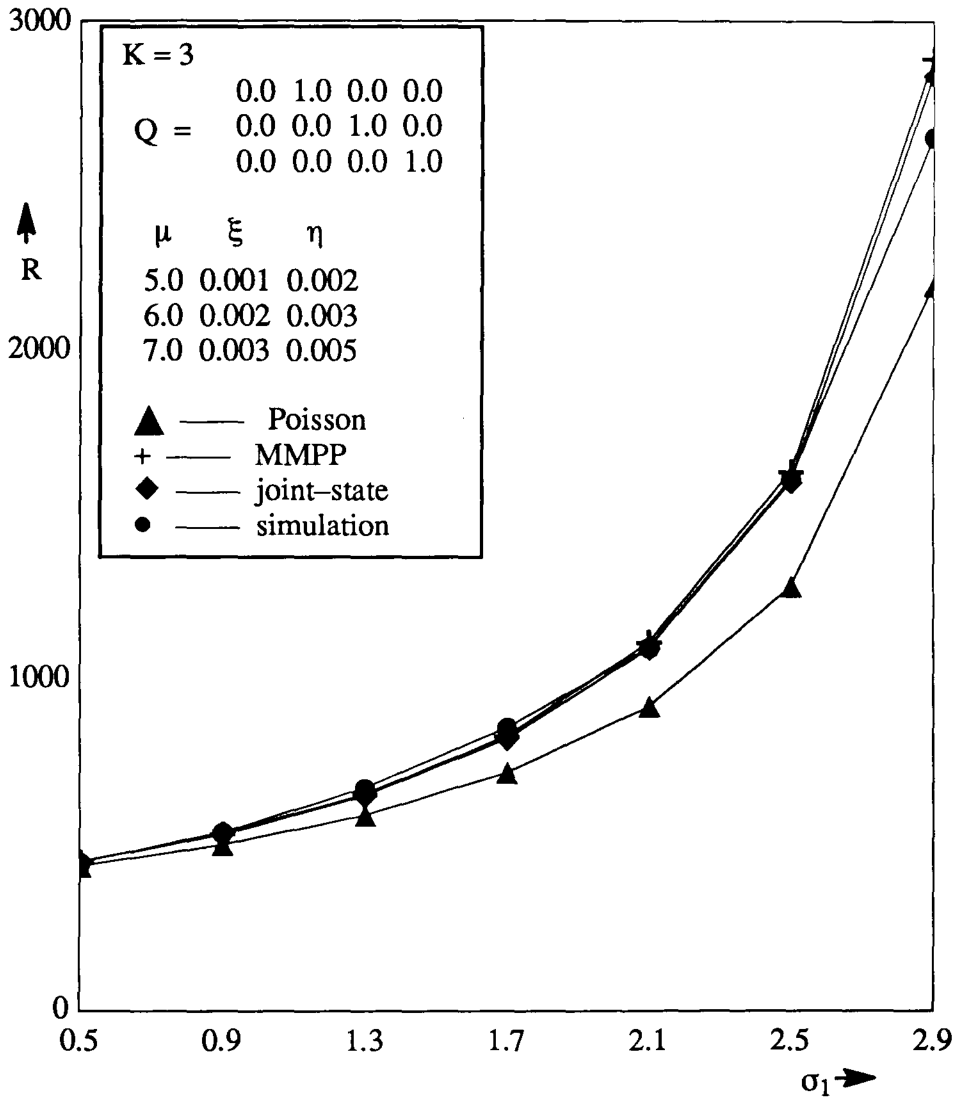

Figure 8.7: 3-node networks: Example 1.

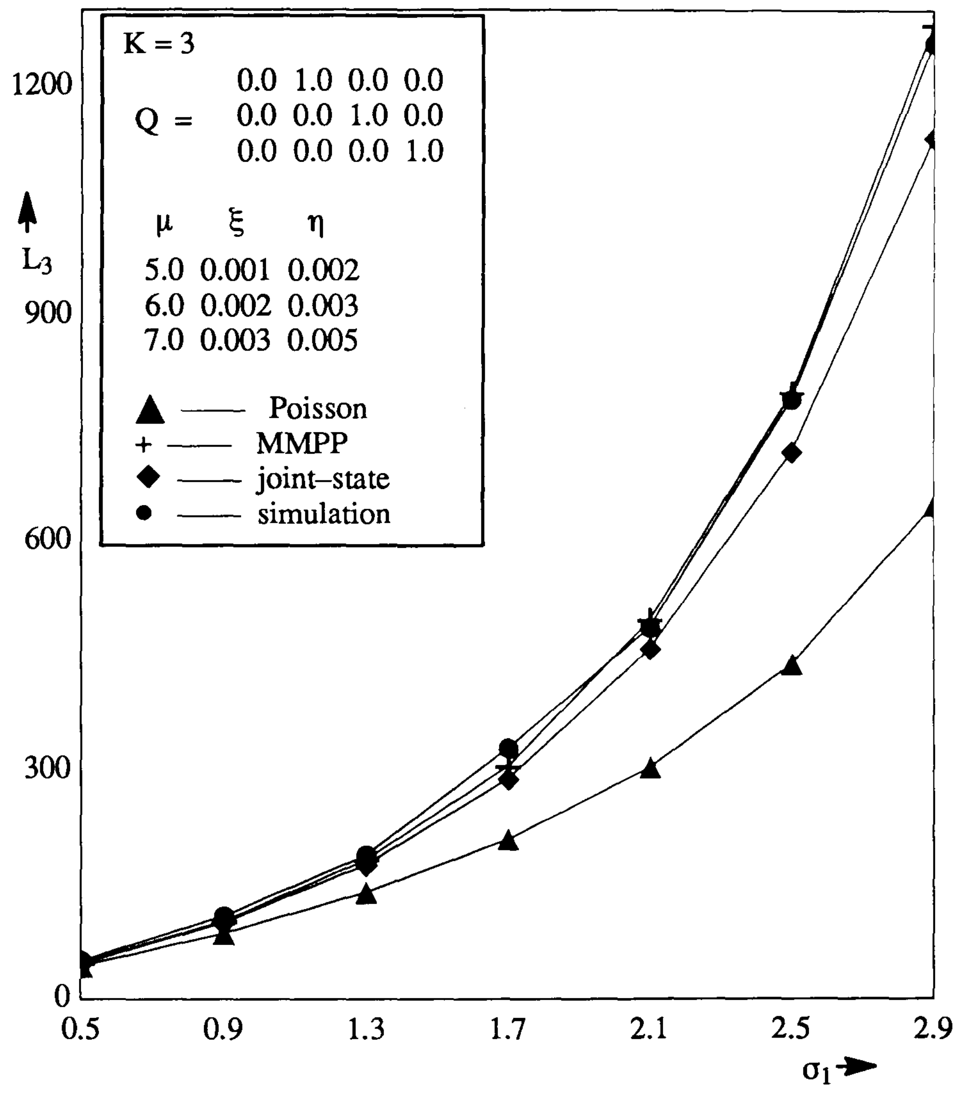

Figure 8.8: 3-node networks: Example 1.

With the same 3-nodes as in the previous example, but with a different topology, another network, i.e. Example 2, is considered. The results are shown in Figures 8.9 and 8.10. Response time vs. σ_1 is plotted in Figure 8.9, L_1 vs. σ_1 in Figure 8.10. Here, L_1 is taken for plotting because, that seems to have least accuracy in the model computations. Table 8.7 displays the performance measures for some values of σ_1. Interestingly, the Joint-state model does very well, whereas the results of the MMPP model are not that accurate. It is not difficult to explain, why that is so. For example, in the case of L_1, there are two streams to node 1. One is the external Poisson arrivals (with rate σ_1), the other one is $IPP_{3,1}$. In the model, these two are assumed independent. But actually they are correlated. That is the reason for the inaccuracy in the computation of L_1 by the MMPP model.

In the next example, we further modify the 3-server network, by decreasing the failure rates of servers by a factor of 10. The resulting network is Example 3. The results of this are shown in Figures 8.11 and 8.12, and in Table 8.7. Here too, the Joint-state model performs well. The MMPP model and the Poisson approximation do not perform very well.

Two example networks with $K = 4$, are taken up. Example 1 is a tandem network. Its parameters and results are displayed in Figures 8.13, 8.14, and in Table 8.7. σ_2, σ_3 and σ_4 are taken as zeroes. In the figures, R and L_4 are plotted against σ_1. The MMPP model and the Joint-state model have given good results.

The last example, i.e. Example 2 for $K = 4$, is shown in Figures 8.15, 8.16 and in Table 8.7. The parameters of the networks are shown in the figures. Again, only node 1 has external arrivals. R and L_4 are plotted against σ_1. The MMPP model and the Joint-state model do well.

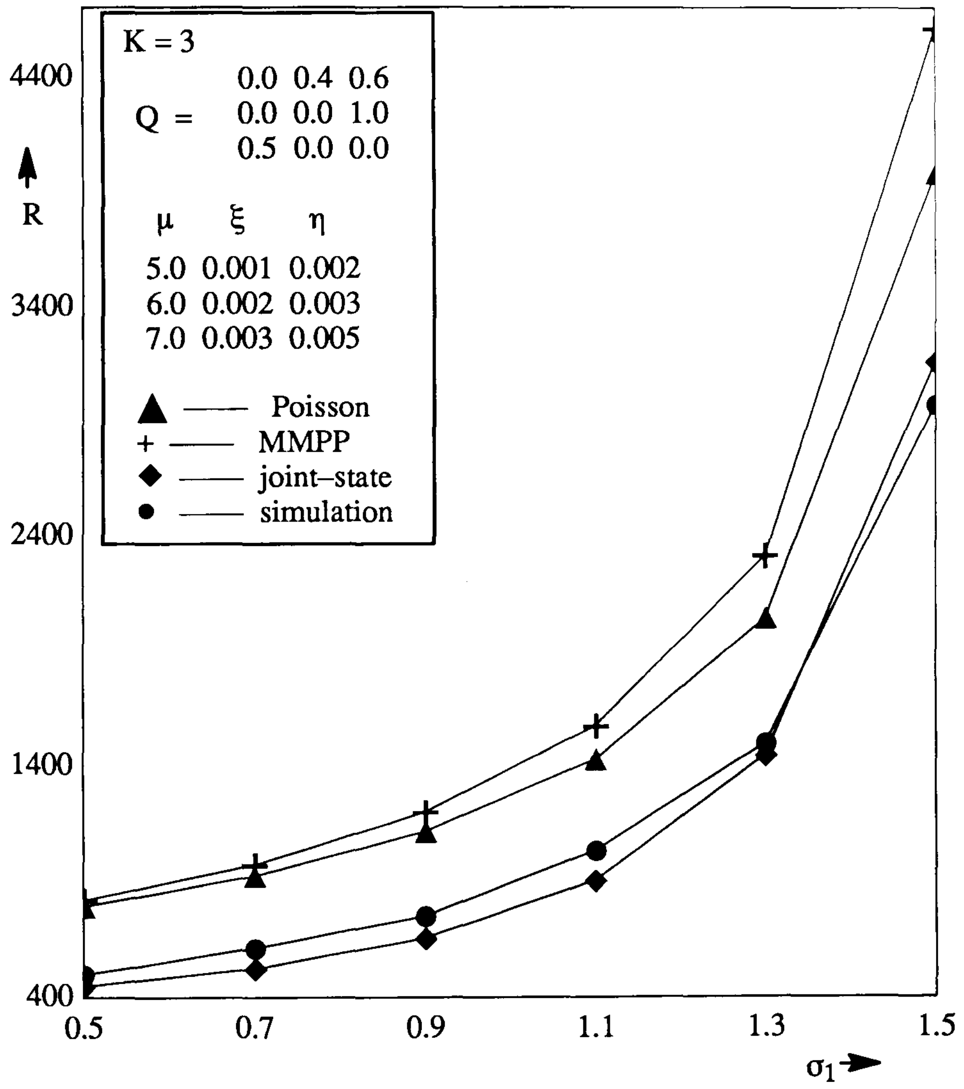

Figure 8.9: 3-node network: Example 2.

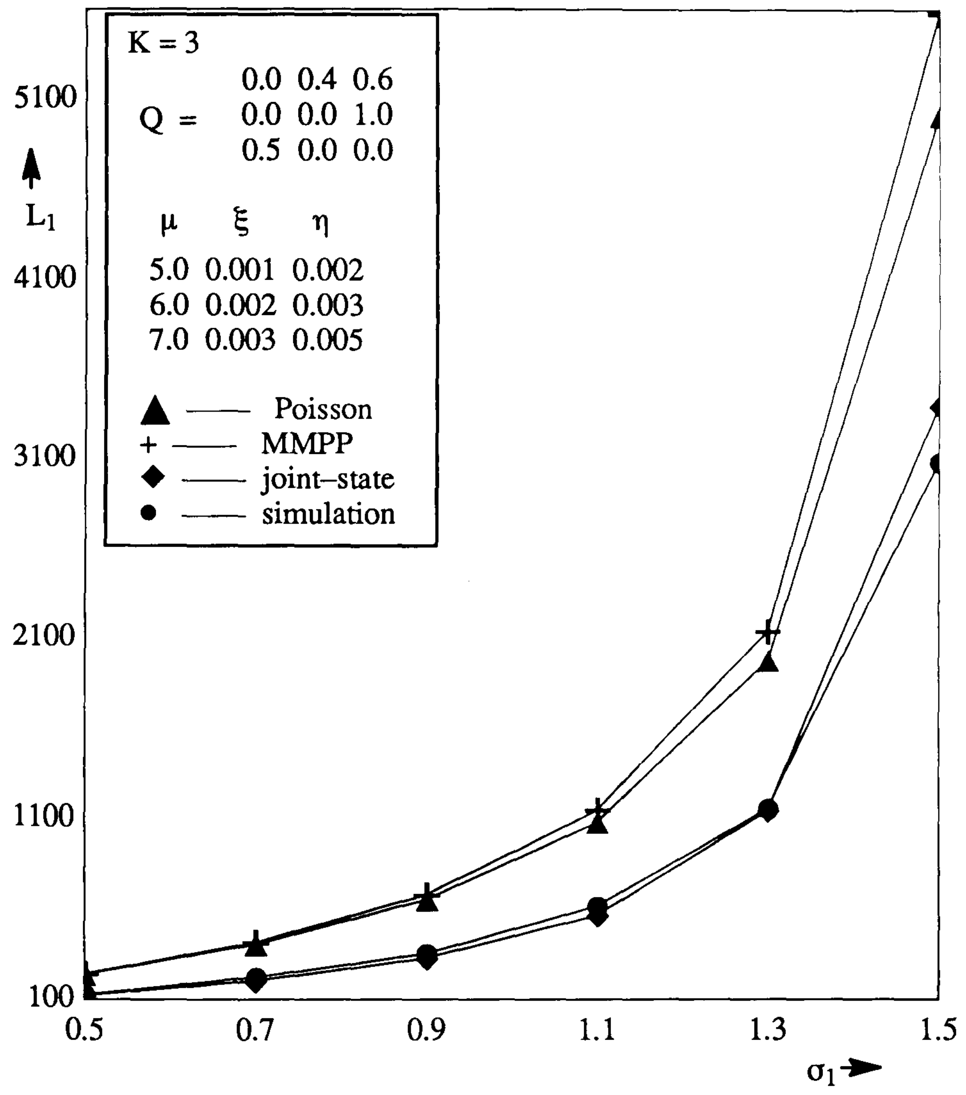

Figure 8.10: 3-node network: Example 2.

Table 8.2: 3-node network: Example 2.

$\sigma_1 = 0.5$				
P.M.s	Poisson	MMPP	Joint-state	Simulation
$L_1 =$	238.5	243.0	121.8	127.5
$L_2 =$	60.1	62.0	47.3	50.1
$L_3 =$	97.5	104.4	54.5	71.2
$R =$	792.3	818.7	447.0	497.6
$\sigma_1 = 0.9$				
P.M.	Poisson	MMPP	Joint-state	Simulation
$L_1 =$	653.3	681.4	327.5	353.3
$L_2 =$	120.2	127.8	104.8	110.5
$L_3 =$	230.1	269.2	157.8	211.9
$R =$	1115.2	1198.2	655.7	750.7
$\sigma_1 = 1.3$				
P.M.	Poisson	MMPP	Joint-state	Simulation
$L_1 =$	1973.2	2138.1	1143.9	1155.7
$L_2 =$	195.4	215.1	200.4	199.8
$L_3 =$	482.1	649.3	531.9	587.8
$R =$	2039.0	2309.7	1443.2	1494.7

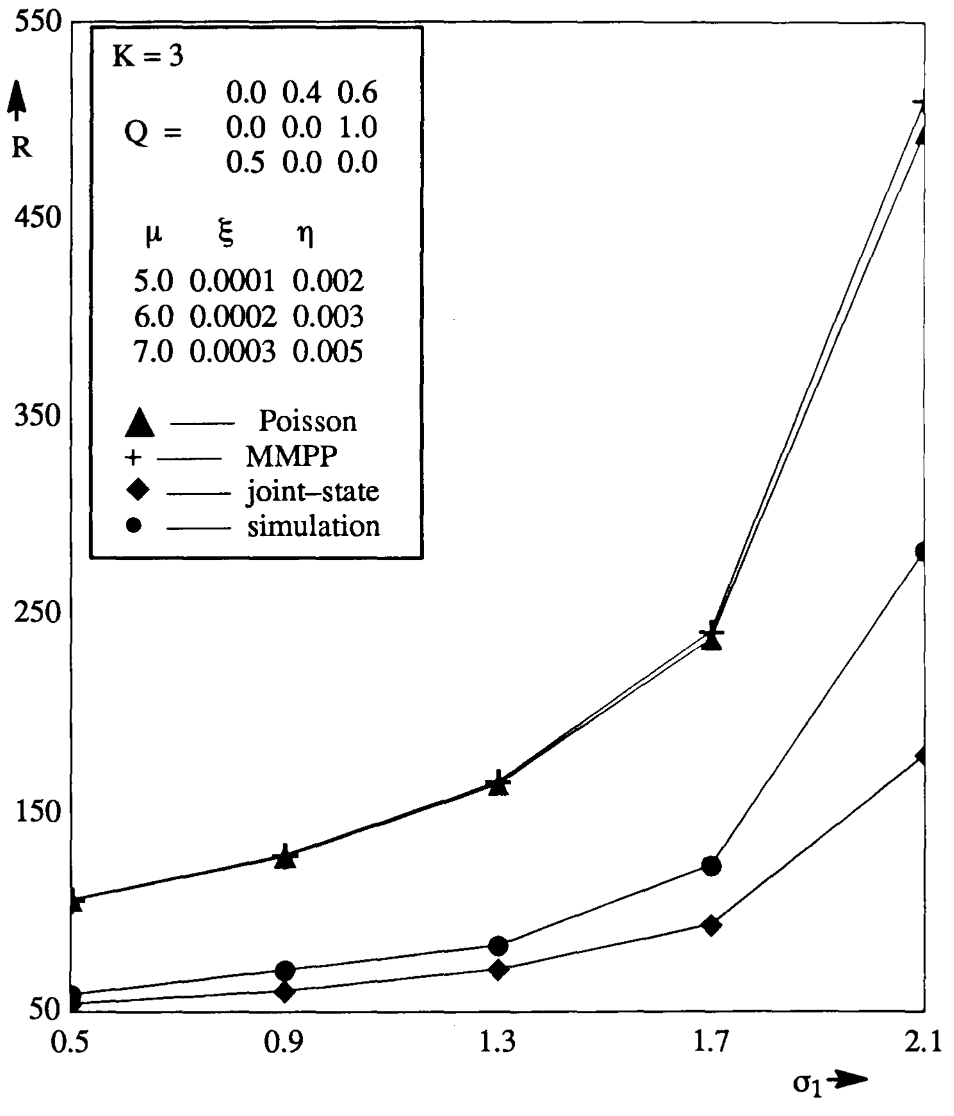

Figure 8.11: 3-node network: Example 3.

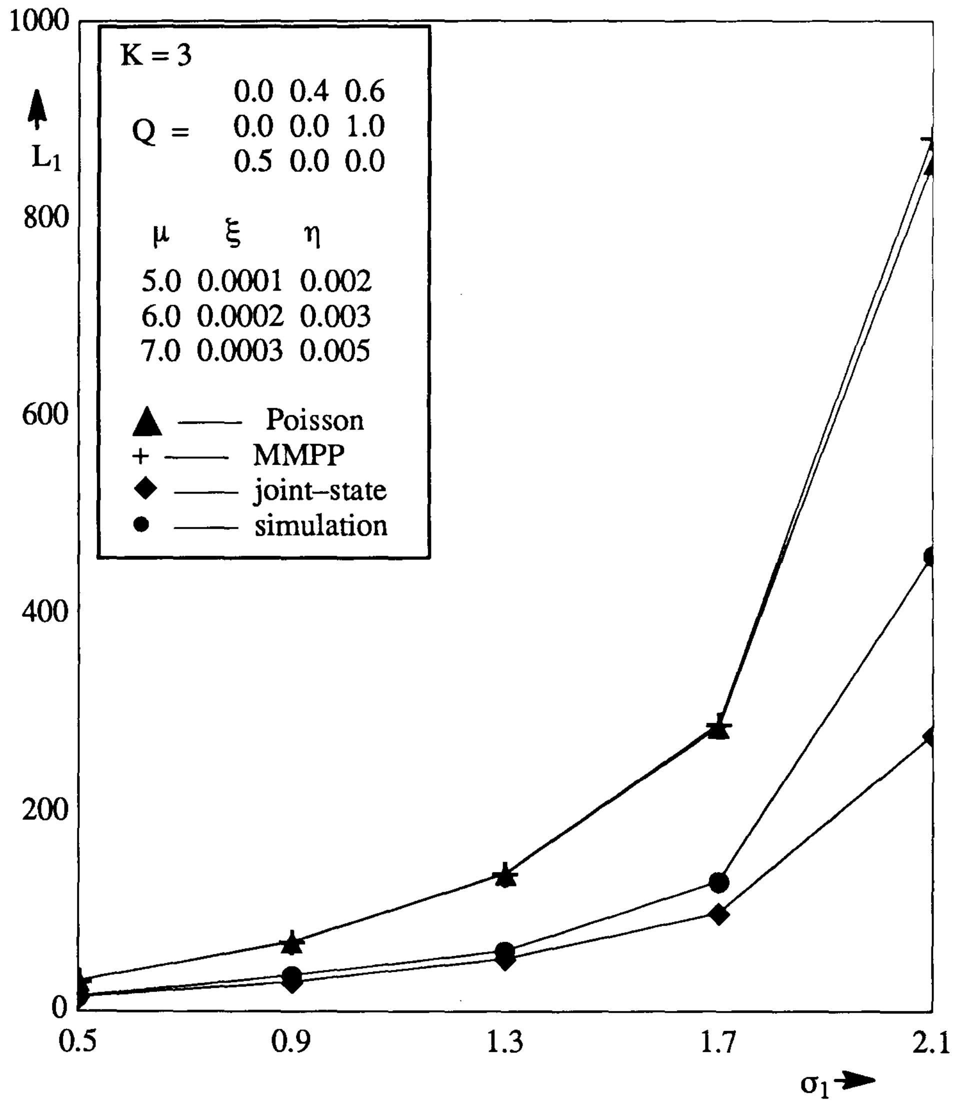

Figure 8.12: 3-node network: Example 3.

Table 8.3: 3-node network: Example 3.

$\sigma_1 = 0.5$				
P.M.s	Poisson	MMPP	Joint-state	Simulation
$L_1 =$	30.4	30.4	14.0	15.0
$L_2 =$	9.0	9.1	6.4	6.9
$L_3 =$	13.5	13.6	6.4	7.3
$R =$	105.9	106.2	53.8	58.4
$\sigma_1 = 1.3$				
P.M.	Poisson	MMPP	Joint-state	Simulation
$L_1 =$	137.6	138.4	52.5	60.5
$L_2 =$	26.8	27.0	18.7	18.6
$L_3 =$	49.2	50.1	21.3	29.1
$R =$	164.3	165.7	71.1	83.2
$\sigma_1 = 2.1$				
P.M.	Poisson	MMPP	Joint-state	Simulation
$L_1 =$	854.9	881.5	277.5	458.7
$L_2 =$	50.3	51.0	37.1	39.5
$L_3 =$	132.4	139.8	61.0	94.1
$R =$	494.1	510.6	178.9	282.0

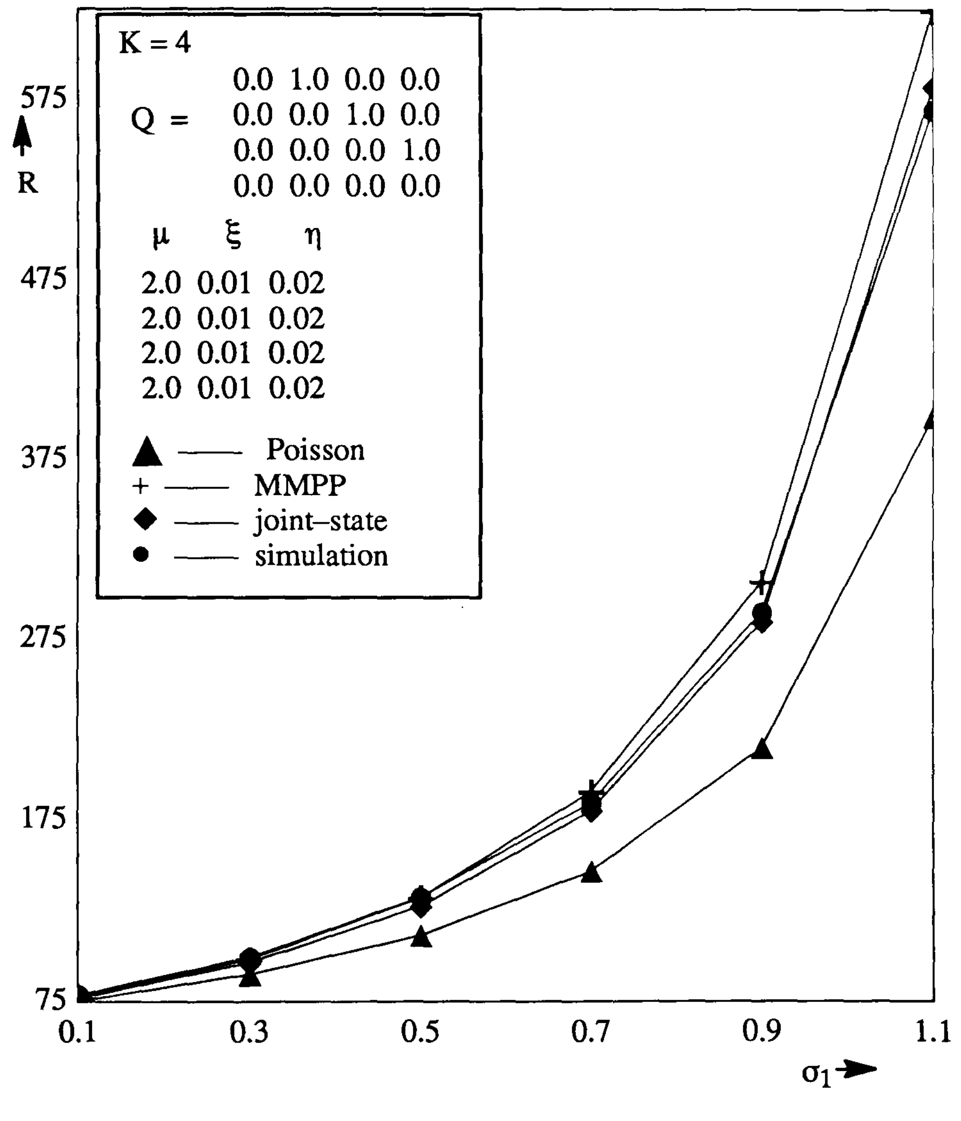

Figure 8.13: 4-node network: Example 1.

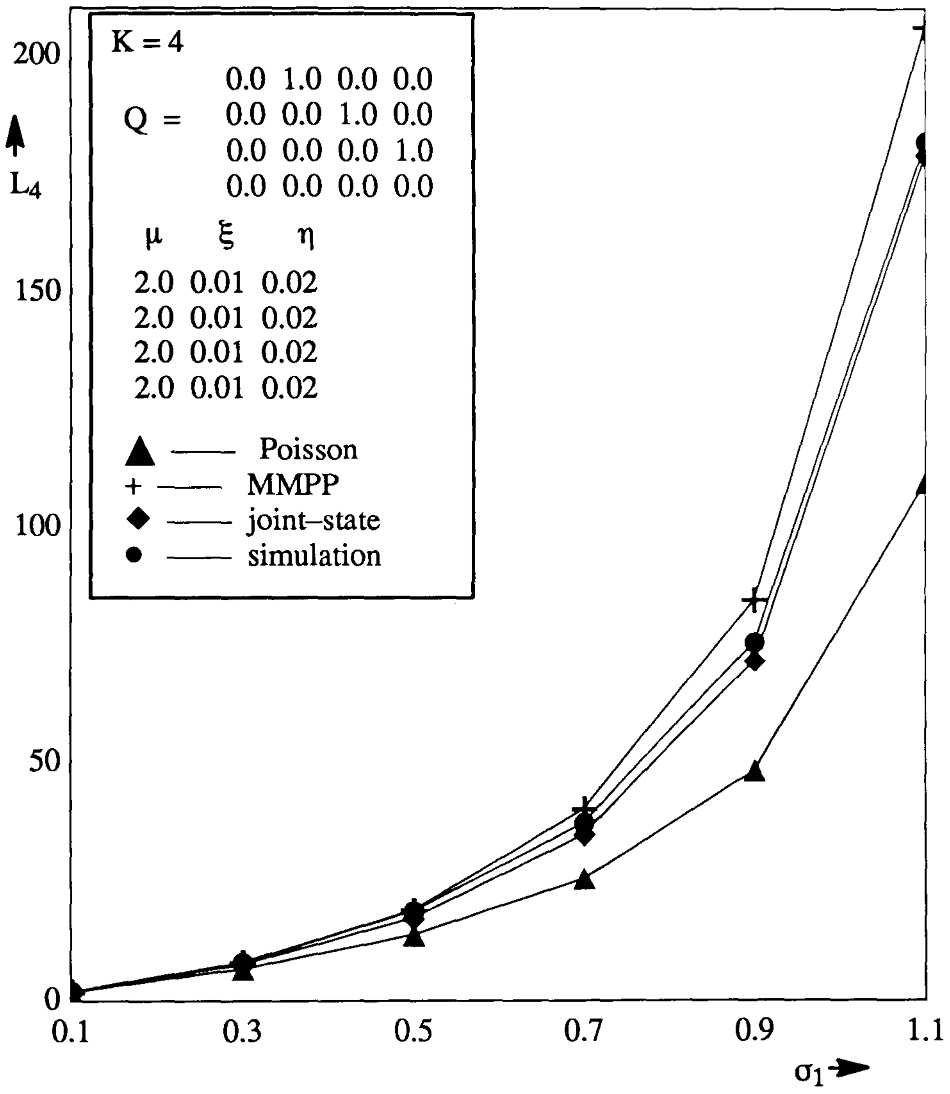

Figure 8.14: 4-node network: Example 1.

Table 8.4: 4-node network: Example 1.

$\sigma_1 = 0.3$				
P.M.s	Poisson	MMPP	Joint-state	Simulation
$L_1 =$	6.7	6.7	6.7	6.7
$L_2 =$	6.7	7.1	7.1	7.3
$L_3 =$	6.7	7.6	7.4	7.7
$L_4 =$	6.7	8.1	7.7	8.1
$R =$	89.9	98.5	96.6	99.5
$\sigma_1 = 0.7$				
P.M.	Poisson	MMPP	Joint-state	Simulation
$L_1 =$	25.7	25.7	25.7	25.6
$L_2 =$	25.7	31.0	31.0	31.2
$L_3 =$	25.7	36.3	34.1	34.8
$L_4 =$	25.7	40.3	35.0	37.4
$R =$	146.7	190.4	179.7	184.2
$\sigma_1 = 1.1$				
P.M.	Poisson	MMPP	Joint-state	Simulation
$L_1 =$	109.5	109.5	109.5	112.9
$L_2 =$	109.5	171.5	171.4	152.2
$L_3 =$	109.5	199.6	179.7	178.9
$L_4 =$	109.5	206.2	179.3	182.0
$R =$	398.1	624.3	581.7	569.0

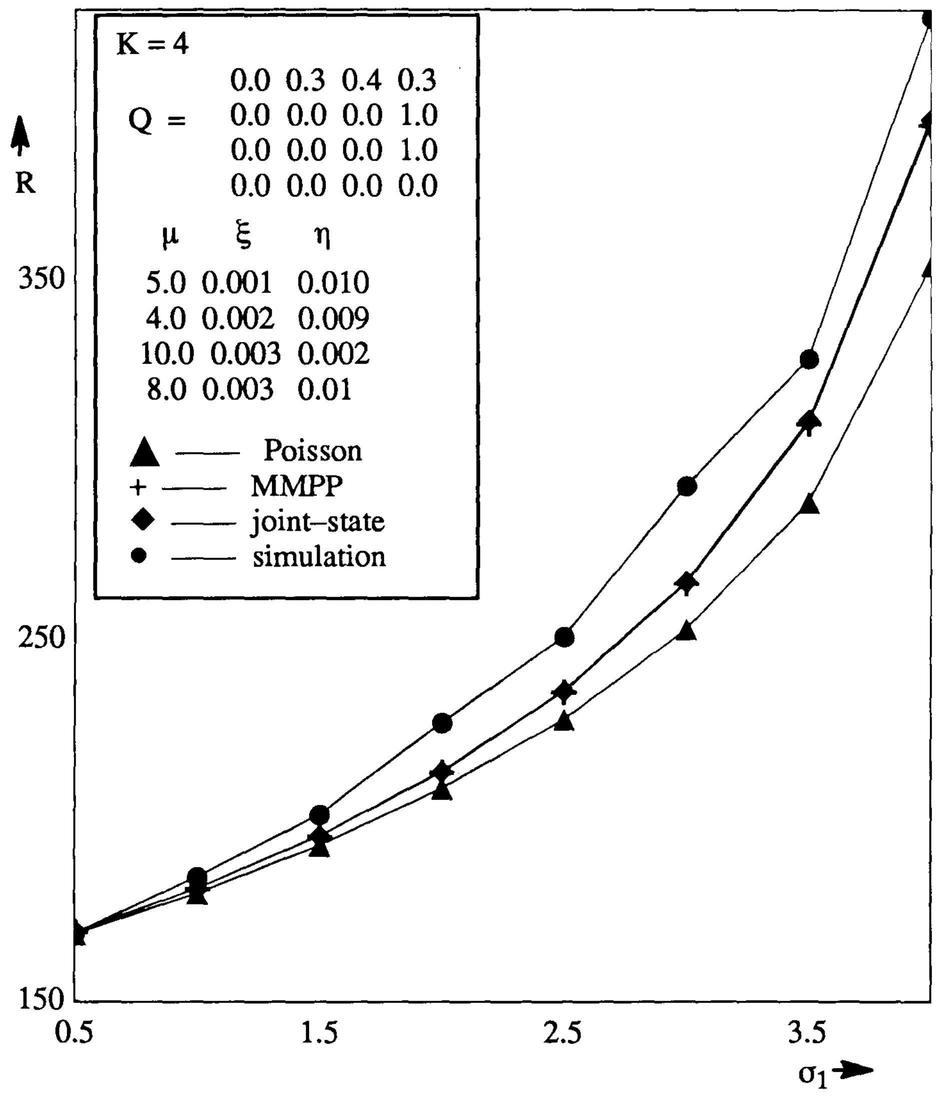

Figure 8.15: 4-node network: Example 2.

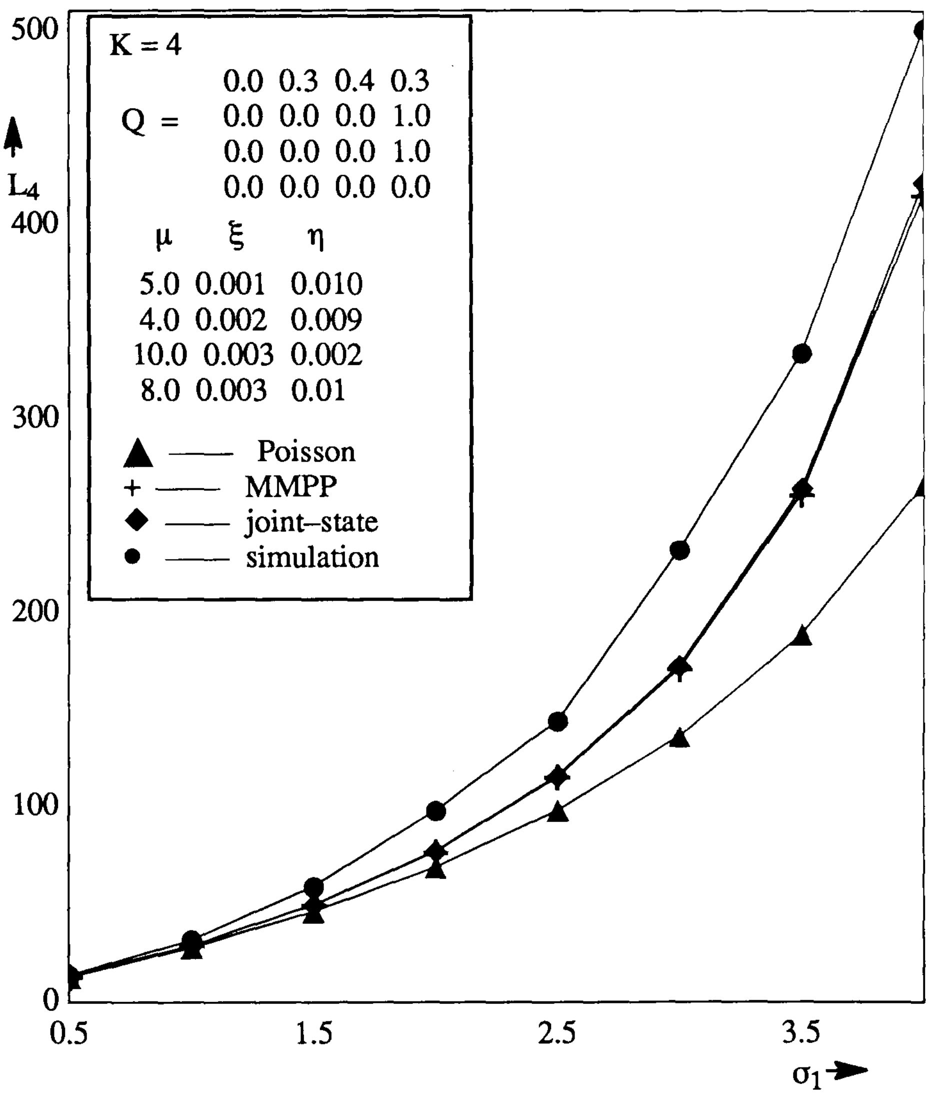

Figure 8.16: 4-node network: Example 2.

Table 8.5: 4-node network: Example 2.

$\sigma_1 = 1.0$				
P.M.s	Poisson	MMPP	Joint-state	Simulation
$L_1 =$	11.9	11.9	11.9	12.2
$L_2 =$	6.8	6.8	6.8	6.9
$L_3 =$	133.4	133.7	133.7	133.9
$L_4 =$	27.7	28.9	28.9	31.6
$R =$	179.9	181.3	181.3	184.5
$\sigma_1 = 2.5$				
P.M.	Poisson	MMPP	Joint-state	Simulation
$L_1 =$	51.7	51.7	51.7	52.7
$L_2 =$	20.0	20.2	20.2	20.2
$L_3 =$	400.3	402.1	402.1	410.0
$L_4 =$	97.9	114.9	115.6	144.1
$R =$	227.9	235.6	235.9	250.8
$\sigma_1 = 4.0$				
P.M.	Poisson	MMPP	Joint-state	Simulation
$L_1 =$	310.4	310.4	310.4	325.5
$L_2 =$	38.9	39.8	39.8	39.8
$L_3 =$	800.7	806.8	806.8	826.0
$L_4 =$	265.6	414.9	421.3	500.9
$R =$	353.9	393.0	394.6	423.0

8.8 Conclusions

We have modelled a very complex problem: open queueing networks with server breakdowns and repairs, and with external job arrivals. This is an important problem occuring in computer and communication networks. There has not been any satisfactory solution, so far. Though we could not succeed to bring out a desirable exact solution, we came up with certain useful approximate models that can be solved by numerical algorithms. Spectral expansion is again made use of extensively. Four different models are suggested.

The first, and the most simply thought of, model is the reduced work rate (RWR) approximation. Though the RWR model has been shown to give fairly accurate results in the case of closed queueing networks with unreliable servers, it fails to model the present problem. The results, when compared to long simulation runs, are found to be very inaccurate, normally. The model, however, is applicable if the repair rates are faster than the service rates and job arrival rates, by orders of magnitude. Hence, the RWR model is not a useful one.

The second, is the Poisson approximation. This is shown to be quite accurate in the case of large networks. In small networks, however, if the node to node transition probabilities are of much smaller magnitude, compared to the exit probabilities, Poisson approximation can give good results.

Two approximate models, the MMPP model and the Joint-state model, are suggested for small networks, i.e. for networks with small number of servers. These two approximations are evaluated, using several sample networks. The Joint-state model seems to be satisfactory in most cases. The MMPP model does work well in some cases.

For simulation, the *batch means* method was used in the calculation of the confidence intervals. The total number of job arrivals in each simulation was 10^7. In most of the experiments, the joint-state approximation was within the 90% confidence interval.

Much more elaborate study would be required to point out which model would work better, and in a given situation or network. We leave this problem, as an interesting open problem for the reseach community.

As we have seen, the joint-states are 2^K. That is quite limiting, to the applicability of the model. However, an efficient improvement is possible in the Joint-state model. We call this, *modified* Joint-state model. It is as follows.

Let $G^*(k)$ be the set of nodes such that, if a job after its service at node l ($l = 1, 2, \ldots, K\,;\, l \neq k$), has a non-zero probability of visiting node k before it leaves the network, then node $l \in G^*(k)$, otherwise node $l \notin G^*(k)$. Obviously, $G(k) \subseteq G^*(k)$. Let the cardinality of $G^*(k)$ be g_k^*. Then, $K - 1 \geq g_k^* \geq g_k$. With these definitions, it is possible to model node k ($k = 1, 2, \ldots, K$), as a Markov-modulated queue with $2^{g_k^*+1}$ arrival phases. These $2^{g_k^*+1}$ arrival phases correspond to the joint-states of all the nodes in $G^*(k)$, plus node k. Hence, in this model, the number of arrival phases of each of the nodes, can be different. If $g_k^* < K - 1$ for any of the nodes, then the computation time needed here would be less than that of the Joint-state model, considered earlier. However, we have not implemented this model.

8.9 Contributions

The contribution of this chapter is,

- A complex modelling problem, open queueing networks with breakdowns and repairs, is considered. Four approximate models are suggested. The accuracy of these models is evaluated by simulation, using several sample networks. These models have different accuracy-efficiency trade-offs.

This work is presented in our paper [10].

Chapter 9

Conclusions

Finally, we intend to list the contributions of this thesis, and go on to describe some research topics that arise for further investigation. Section 9.1 deals with the former, and section 9.2 with the latter.

9.1 Contributions

In each chapter, the contributions have clearly been stated. Many of these contributions are either already published, or accepted for publication, in leading journals and conferences. References to these publications are also provided. A brief summary of these contributions is, as below.

- The Markov processes X, Y and Z are defined. Starting from the spectral expansion solution for the stationary probability invariant vector of the QBD process X, an efficient algorithmic solution is developed. The spectral expansion solution of the process Y, and of the QBD process with *boundary*, Z, are developed. Efficient computational algorithms for these two, are also devised. The necessary stability and spectral analyses of the processes are carried out. This is done in Chapters 2 & 4.

- For the process X, a comparative study of the solution algorithms of the spectral expansion and that of the matrix-geometric method, is performed. This is done using a popular non-trivial example, the modelling of homogeneous multiprocessor systems with general breakdowns and repairs. The results are in favour of the spectral expansion method. Relative computation times are also given. This is done in Chapter 5.

A number of complex modelling problems are solved, for steady state performance and dependability. They are,

- A homogeneous multiprocessor system, with unbounded queueing capacity, and with general breakdowns and repairs of processors, is considered. The system is modelled by the process X, and solved by spectral expansion. Steady state probabilities and performability measures are derived. Numerical examples of typical systems are dealt, and the numerical results are shown in figures. This is done in Chapter 3.

- A serial transfer line with two servers and a buffer in-between, is analysed for steady state performance using the spectral exapansion method. Results of numerical examples are displayed in figures. This is done in Chapter 3.

- A homogeneous multiprocessor, with finite job capacity and with general breakdowns and repairs of processors is considered. The system is modelled by the process Z and sloved by spectral expansion. Some numerical results are presented via graphs. This is in Chapter 4.

- A heretogeneous multiprocessor with unbounded job queue is taken up. The processors are prone to independent breakdowns and repairs. This system is modelled by the process X, and solved by spectral expansion. The importance of the repair policy is highlighted. In some special cases, optimal repair strategies are worked out. In the general case, some nearly optimal *heuristics* for repair strategy are suggested. These heuristics are evaluated by an extensive simulation study. Numerical results, through example systems, are presented. This is done in Chapter 6.

- Due to the occurrence of bursty arrivals in practical systems, an important queue is taken up. This is an exponential server with breakdowns and Markov-modulated arrivals, and with unbounded queueing capacity. The system is modelled by the process X, and solved by spectral expansion. Numerical performance results, on example systems, are presented. the necessity of modelling the bursty process is explained, through these results. An important and interesting problem is to develop a computationally simple and effective model for the departure process. That is done, by modelling the departure process as an interrupted Poisson process. That model is validated through examples. All this is done in Chapter 7.

- A very complex modelling problem, that of open queueing networks with breakdowns and repairs is considered. Four different approximate models are conceived. They are, the RWR approximation, the Poisson approximation, the MMPP model and the Joint-state model. The last two are iterative, and spectral expansion is extensively used in these. These models differ in their accuracy-efficiency trade-off. The RWR model has a very limited applicability. The Poisson approximation is shown to be applicable in large networks. The MMPP model and the Joint-state model are suitable for small networks. A number of numerical examples are worked out. The MMPP model works moderately well, and the Joint-state model works quite well. This is done in Chapter 8.

Necessary discussion is presented in each chapter, in the section on conclusions.

9.2 Open problems

A number of open problems arise from this research. Some of them are already briefed in each chapter, in the section on conclusions. They are,

- We feel it may be possible to prove the stability condition, 2.21 and 2.23, mathematically. We are already working on it. In the case of process Y also, we are working on a mathematical proof for the stability condition.

- We have written our programs in SIMULA. For the eigenvalue problem and for the solution of linear simultaneous equations, we have resorted to the NAG routines. When the number of operative states, $N + 1$, is large, we noticed some inaccuracies in the NAG results. For example, the eigenvalue at 1.0 is computed with considerable inaccuracy. Also, the LU-decomposition routine for the solution of simultaneous equations has not been robust enough, for large $N + 1$. A solution for this problem would be to develop self contained programs for spectral expansion, say in C++, with highest possible precision, without resorting to NAG routines, and using accurate and robust algorithms. Also, since the matrices in equations (2.28) and (2.31) are sparse, that may lead to some computational saving as well.

- The M/G/1 type of Markov processes [52] is a subset of the process Y. It may be posible to simplify further, the spectral expansion solution of such processes. Our ideas are too preliminary to explain further.

- There have been some very recent improvements in the matrix-geometric solution methodology [65]. It is worth comparing this improved methodology with spectral expansion, for computational efficiency.

- The analysis of the multi-server delay-loss queue, $\sum_i MAP_i/Ph/K/L$ is an important model, used in B-ISDN networks. This sytem can be modelled by the process Z and can be analysed by the methodology given in Chapter 4. For such a queue, if $K = 1$, we feel the departure process can well be modelled as an interrupted Poisson process, following the analysis in Chapter 7. This can be done, when L is finite or infinite. If $K > 1$, then an appropriate model for the departure process may be an MMPP with $K + 1$ phases. Such study can be interesting and useful.

- In the case of open queueing networks with breakdowns, the modified Joint-state method is briefly described in Chapter 8. This may be a promising method, and can be computationally very effective if the transition rate matrix Q is sparse. In such a case of sparse Q, the method can be applicable even for large networks, achieving good accuracy. Also, it is worth to check if the methodology outlined in this chapter, can be extended, with necessary modifications, to include, joint-state dependent routing, and also to loss networks.

epilogue

Research in Markov modelling is important, and it has a lot more role to play in future, in the analysis of discrete event systems. We hope this thesis would serve to move that research a gentle step forward.

Lastly, we conclude this thesis, with a sincere heart-felt wish that this research work would benefit mankind, in some way.

Bibliography

[1] B. Avi-Itzhak and P. Noar, **Some queueing problems with the service station subject to breakdowns**, *Operations Research*, Vol. 11, No. 3, pp. 303-320, May 1963.

[2] F. Baskett, K.M. Chandy, R.R. Muntz and F. Palacios-Gomez, **Open, closed and mixed networks of queues with different classes of customers**, *Journal of the ACM*, Vol. 22, No. 2, pp. 248-260, April 1975.

[3] S.E. Butner and R.K. Iyer, **A statistical study of reliability and system load at SLAC**, *Proceedings of FTCS-10*, pp. 207-209, October 1980.

[4] R. Chakka and I. Mitrani, **Multiprocessor systems with general breakdowns and repairs**, Extended Abstract, *Performance Evaluation Review*, Special Issue, Vol. 20, No. 1, pp. 245-246, June 1992.

[5] R. Chakka and I. Mitrani, **A numerical solution method for multiprocessor systems with general breakdowns and repairs**, *Procs., 6th Int. Conf. on Performance Tools and Techniques*, Edinburgh, pp. 289-304, September 1992.

[6] R. Chakka and I. Mitrani, **Heterogeneous multiprocessor systems with breakdowns - performance and optimal repair strategies**, Presented

at the *INRIA/ ORSA/ TIMS/ SMAI Conference on Applied Probability in Engineering, Computer and Communiation Sciences*, Paris, June 1993.

[7] R. Chakka and I. Mitrani, **Heterogeneous multiprocessor systems with breakdowns: performance and optimal repair strategies**, *Theoretical Computer Science*, Vol. 125, No. 1, pp. 91-109, March 1994.

[8] R. Chakka and I. Mitrani, **Fast numerical solution for a class of Markov models**, in *Predictably Dependable Computing Systems*, (eds. B. Randell etal), Springer Verlag, pp. 519-534, June 1995.

[9] R. Chakka, **Spectral expansion solution for a finite capacity multiserver system in a Markovian environment**, *Proceedings of the Third International Workshop on Queueing Networks with Finite Capacity*, Bradford, pp. 6/1-6/9, July 1995.

[10] R. Chakka and I. Mitrani, **Approximate analysis of open queueing networks with breakdowns and repairs**, Presented at *Stochastic Networks Workshop*, Edinburgh, August 1995.

[11] E. Cinlar, **Superposition of point processes**, in *Stochastic Point Processes: Statistical Analysis, Theory and Applications*, (ed. Lewis), Wiley and Sons Inc., New York, pp. 549-606, 1972.

[12] J.N. Daigle and D.M. Lucantoni, **Queueing systems having phase dependent arrival and service rates**, in *Numerical Solutions of Markov Chains*, (ed. W.J. Stewart), Marcel Dekker, New York, pp. 161-202, 1991.

[13] A.I. Elwalid, D. Mitra and T.E. Stern **Statistical multiplexing of Markov modulated sources: theory and computational algorithms**, in *Teletraffic and Data Traffic in a Period of Change*, (eds. A. Jenson and V.B. Iversen), *International Teletraffic Congress-13*, Copenhagen, pp. 495-500, June 1991.

[14] M. Ettl and I. Mitrani, **Applying spectral expansion in evaluating the performance of multiprocessor systems**, *Proceedings of the 3rd QMIPS Workshop: Part 1*, (eds. O.J.Boxma and G.M.Koole), CWI TRACT, Amsterdam, pp. 45-58, 1994.

[15] R.V. Evans, **Geometric distribution in some two dimensional queueing systems**, *Operations Research*, Vol. 15, No. 5, pp. 830-846, September 1967.

[16] W. Fischer and K. Meier-Hellstern, **The Markov-modulated Poisson process**, *Performance Evaluation*, Vol. 18, No. 2, pp. 149-171, September 1993.

[17] H.R. Gail, S.L. Hantler and B.A. Taylor, **Spectral analysis of M/G/1 type Markov chains**, RC17765, IBM Research Division, 1992.

[18] F.R. Gantmacher, **The Theory of Matrices**, Vol. 1 & 2, *Chelsea Publishing Company*, New York, 1959.

[19] D.P. Gaver, **A waiting line with interrupted service including priorities**, *Journal of the Royal Statistical Society Ser. B*, Vol. 24, pp. 73-90, 1962.

[20] E. Gelenbe and I. Mitrani, **Analysis and Synthesis of Computer Systems**, *Academic Press*, New York, 1980.

[21] I. Gohberg, P. Lancaster and L. Rodman, **Matrix Polynomials**, *Academic Press*, New York, 1982.

[22] G.H. Golub and C.F Van Loan, **Matrix Computations**, 2nd edition, *The John Hopkins University Press*, 1989.

[23] A. Goyal, P. Shahabuddin, P. Heidelberger, V.F. Nicola and P.W. Glynn, **A unified framework for simulating Markovian models of highly dependable systems**, *IEEE Transactions on Computers*, Vol. 41, No. 1, pp. 36-51, January 1992.

[24] A.G. Greenberg, B.D. Lubachevsky and I. Mitrani, **Algorithms for unboundedly parallel simulations**, *ACM Transactions on Computer Systems*, Vol. 9, No. 3, pp. 201-221, 1991.

[25] L. Gun, **Experimental results on matrix-analytical solution techniques - extensions and comparisons**, *Stochastic Models*, Vol. 5, No. 4, 1989.

[26] H. Heffes and D.M. Lucantoni, **A Markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance**, *IEEE Journal on Selected Areas of Communications*, Vol. 4, No. 6, pp. 856-868, 1986.

[27] Y.C. Ho, **Perturbation analysis : concepts and algorithms**, in *Proc. of the Winter Simulation Conference*, pp. 231-239, 1992.

[28] I. Ide, **Superposition of interrupted Poisson processes and its application to packetized voice multiplexers**, in: *Teletraffic Science for New Cost-Effective Systems, Networks and Services* (ed. M. Bonatti), *Proc. 12th International Teletraffic Congress (ITC-12)*, Torino, Italy, pp. 1399-1405, June 1988.

[29] R.K. Iyer, D.J. Rosetti and M.C. Hsueh, **Measurement and modeling of computer reliability as affected by system activity**, *ACM Transactions Computer Systems*, Vol.4, No.3, pp. 214-237, August 1986.

[30] J.R. Jackson, **Networks of waiting lines**, *Operations Research*, Vol. 5, No. 4, pp. 518-521, August 1957.

[31] A. Jennings, **Matrix Computations for Engineers and Scientists**, *John Wiley and Sons*, New York, 1977.

[32] H. Kameda, **A finite-source queue with different customers**, *Journal of the ACM*, Vol. 29, No. 2, pp. 478-491, April 1982.

[33] H. Kameda, **Realizable performance vectors of a finite source queue**, *Operations Research*, Vol. 32, No. 6, pp. 1358-1367, November 1984.

[34] J. Keilson, U. Sumita and M. Zachmann, **Row-continuous finite Markov chains: structure and algorithms**, *Journal of the Operations Research Society of Japan*, Vol. 30, No. 3, pp. 291-314, September 1987.

[35] A. G. Konheim and M. Resier, **A queueing model with finite waiting room and blocking**, *Journal of the ACM*, Vol. 23, No. 2, pp. 328-341, April 1976.

[36] G. Koole and M. Vrijenhoek, **Scheduling a repairman in a finite source system**, Working Report: TW-95-03, Department of Mathematics and Computer Science, Leiden University, Leiden, 1995.

[37] P. Lancaster, **Theory of Matrices**, *Academic Press*, New York, 1969.

[38] G. Latouche, P.A. Jacobs and D.P. Gaver, **Finite Markov chain models skip-free in one direction**, *Naval Research Logistics Quarterly*, Vol. 31, No. 4, pp. 571-588, December 1984.

[39] T. Lehtonen, **Stochastic comparisons for many server queues with non-homogeneous exponential servers**, *OPSEARCH*, Vol. 20, No. 1, pp. 1-15, 1983.

[40] D.M. Lucantoni, K.S. Meier-Hellstern and M.F. Neuts, **A single-server queue with server vacations and a class of non-renewal arrival processes**, *Advances in Applied Probability*, Vol. 22, pp. 676-705, 1990.

[41] K.S. Meier-Hellstern, **The analysis of a queue arising in overflow models**, *IEEE Transactions on Communications*, Vol. 37, No. 4, pp. 367-372, April 1989.

[42] I. Mitrani and B. Avi-Itzhak, **A many-server queue with service interruptions**, *Operations Research*, Vol. 16, No. 3, pp. 628-638, May 1968.

[43] I. Mitrani, **Networks of unreliable computers**, in *Computer Architectures and Networks*, (eds. E. Gelenbe and R. Mahl), North Holland Publishing Company, pp. 359-373, 1974.

[44] I. Mitrani and P.J.B. King, **Multiserver systems subject to breakdowns: An empirical study**, *IEEE Transactions on Computers*, Vol. 32, No. 1, pp. 96-99, January 1983.

[45] I. Mitrani and D. Mitra, **A spectral expansion method for random walks on semi-infinite strips**, *IMACS Symposium on Iterative Methods in Linear Algebra*, Brussels, pp. 141-149, 1991.

[46] I. Mitrani and A. Puhalskii, **Limiting results for multiprocessor systems with breakdowns and repairs**, *Queueing Systems*, Vol. 14, pp. 293-311, 1993.

[47] I. Mitrani and R. Chakka, **Spectral expansion solution for a class of Markov models: Application and Comparison with the Matrix-Geometric Method**, *Performance Evaluation*, Vol. 23, No. 3, pp. 241-260, September 1995.

[48] The Numerical Algorithms Group Limited, **The NAG Fortran Library Manual, Mark 14**, (1st Edition), Oxford, April 1990.

[49] M.F. Neuts, **Two queues in series with a finite intermediate waiting room**, *Journal of Applied Probability*, Vol. 5, pp. 123-142, 1968.

[50] M.F. Neuts and D.M. Lucantoni, **A Markovian queue with N servers subject to breakdowns and repairs**, *Management Science*, Vol. 25, pp. 849-861, 1979.

[51] M.F. Neuts, **Matrix Geometric Solutions in Stochastic Models**, *John Hopkins University Press*, Baltimore, 1981.

[52] M.F. Neuts, **Structured Stochastic Matrices of M/G/1 Type and Their Applications**, *Marcel Dekker*, New York, 1989.

[53] N.U. Prabhu and Y. Zhu, **Markov-modulated queueing systems**, *Queueing Systems*, Vol. 5, No. 1-3, pp. 215-246, 1989.

[54] M. Reiser, **A queueing network analysis of computer communication networks with window flow control**, *IEEE Transactions on Communications*, Vol. 27, pp. 1199-1209, August 1979.

[55] L.P. Seelen **An algorithm for Ph/Ph/c queues**, *European Journal of Operations Research*, Vol. 23, pp. 118-127, 1986.

[56] L.P. Seelen, H.C. Tijms and M.H. Van Hoorn, **Tables for Multiserver Queues**, *North-Holland*, Amsterdam, 1986.

[57] B. Sengupta, **A queue with service interruptions in an alternating Markovian environment**, *Operations Research*, Vol. 38, No. 2, pp. 308-318, March 1990.

[58] P. Shahabuddin, **Simulation and analysis of highly reliable systems**, Ph.D. Dissertation, Department of Operations Research, Stanford University, June 1990.

[59] R. Suri, **Robustness of queueing network formulas**, *Journal of the ACM*, Vol. 30, No. 3, pp. 564-594, July 1983.

[60] L. Takacs, **Introduction to the Theory of Queues**, Oxford University Press, New York, 1962.

[61] D. Tang, **Measurement-based dependability analysis and modeling for multicomputer systems**, Ph.D. Thesis, Center for Reliable and High-Performance Computing, University of Illinois at Urbana-Champaign, October 1992.

[62] K. Thiruvengadam, **Queueing with breakdowns**, *Operations Research*, Vol. 11, No. 1, pp. 62-71, January 1963.

[63] H.C. Tijms and M.C.T. Van De Coevering, **A simple numerical approach for infinite-state Markov chains**, *Probability in Engineering and Information Sciences*, Vol. 5, pp. 285-295, 1991.

[64] B. Vinod and T. Altiok, **Approximating unreliable queueing networks under the assumption of exponentiality**, *Journal of Operations Research Society*, Vol. 37, No. 3, pp. 309-316, 1986.

[65] D. Wagner, V. Nauomov and U. Krieger, **Analysis of a finite capacity multi-server delay-loss system with a general Markovian arrival process**, in *Matrix-Analytic Methods in Stochastic Models*, (eds. A.S. Alfa and S. Chakravarthy), Marcel Dekker, New York, 1995.

[66] V. Wallace, **The solution of quasi birth and death processes arising from multiple access computer systems**, Ph.D. Dissertation, TR No: 07742-6-T, Systems Engineering Laboratory, University of Michigan, 1969.

[67] H.C. White and L.S. Christie, **Queueing with preemptive priorities or with breakdowns**, *Operations Research*, Vol. 6, No. 1, pp. 79-95, January 1958.

[68] S. Wolfram, **MATHEMATICA - A System for Doing Mathematics by Computer**, *Addison-Wesley Publishing Company*, 1988.

[69] U. Yechiali, **A queueing-type birth-and-death process defined on a continuous time Markov chain**, *Operations Reseach*, Vol. 21, No. 2, pp. 604-609, March 1973.