THESIS REPORT
Master's Degree

S Y S T E M S
R E S E A R C H
C E N T E R

Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), the University of Maryland, Harvard University, and Industry

Some Problems in Queueing Systems with Resequencing

by S. Varma
Advisor: A. M. Makowski

M.S. 87-9
Formerly TR 87-192

ABSTRACT

Title of Thesis: Some problems in queueing systems with resequencing

Subir Varma, Master of Science, 1987

Thesis directed by: Armand M. Makowski
Associate Professor
Electrical Engineering Department

The aim of this thesis is to solve some problems associated with queueing systems with resequencing of customers. Resequencing is associated with the presence of some disordering system which operates on an input arrival stream of customers and delays each customer by a random amount, so that they may leave the system in a different order than the one in which they entered it. However, if the constraint that the customers should leave the system in the same order in which they entered it, is imposed, then a customer may have to undergo an additional delay, which is known as resequencing delay. Such situations typically arise due to packet switching in a computer communication network, in an interconnection network of a multi-processor architecture and concurency control schemes of distributed data bases among other places. The presence of resequencing makes the analysis of the queueing system intractable in most cases, and very few analytical results are known about these systems.

Two general representation results for resequencing systems are the principal tools used in the thesis. The first representation gives the delay in a general resequencing system, in a recursive sample path form. It is used to investigate multistage resequencing systems and also to deduce some interesting structural properties of multiple server resequencing systems. It is shown that hop-by-hop

resequencing leads to greater delay compared to end-to-end resequencing in N stage infinite server systems in the sense of strong stochastic ordering. The effect of varying the number of servers on the resequencing delay in a finite server system is investigated for both single stage as well as two stage disordering systems, and also several useful structural properties which can be used to develop bounds for intractable resequencing systems, are identified.

The second representation provides a Markovian state space description for the two server resequencing system with exponential interarrival and service times. The state occupation probabilities are calculated for the $M/M/2/B$ queue with resequencing using this representation. The optimal policy for assigning customers to the two servers in the case when they have different service rates, to minimize the total delay (including the resequencing delay), is investigated using dynamic programming arguments. The optimal policy is shown to be of the threshold type in the number of customers in the main queue buffer and independent of the number of customers in the resequencing buffer.

SOME PROBLEMS IN QUEUEING SYSTEMS WITH RESEQUENCING

by

Subir Varma

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
1987

Advisory Committee:

Associate Professor Armand M. Makowski
Associate Professor Prakash Narayan
Associate Professor Satish Tripathi

TO MY DEAR PARENTS

ACKNOWLEDGMENTS

I would like to acknowledge the constant direction and guidance of my advisor, Dr Armand M. Makowski for which I am greatly indebted. His painstaking scrutiny of the entire thesis helped immeasurably in improving its content as well as its style. I would also like to thank the Systems Research Center at the Univsersity of Maryland for financial support during the period when this thesis was written.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Introduction

This thesis deals with several problems that arise in a queueing system with resequencing of customers. Since these systems can be imbedded in the general framework of systems with synchronization constraints, we provide in this introductory chapter, a short discusion of the more general problem.

This chapter is organized as follows. In Section 1.2 we give a number of examples from computer systems where synchronization constraints are important. We also provide a generic classification of synchronization constraints that covers the cases that were discussed. In Section 1.3, we summarize some known analytical results concerning such systems. A more detailed discussion of the analytical results that are known about resequencing systems is given in Section 1.4. Finally in Section 1.5, we summarize the main results developed in this thesis.

1.2 Examples of Systems with Synchronization Constraints

Systems with synchronization constraints are ubiquitious in modern computer and communication network architectures. The main reason for this can be traced back to the parallel architecture or distributed nature of these systems. Because of parallelism several processes may be running in the system at the same time. However these processes are not independent of each other and they ocassionally need to communicate to exchange data. Concurrent processes can talk to each other properly only when they are properly synchronized with each other and this leads to synchronization constraints.

Parallelism in system architectures has increased dramatically in recent years [67], [70]. One of the reasons for this is the steep fall in the price of the main constituents of a computer, such as the CPU and memory, and it has therefore become economically feasible to design computers with thousands of processing units. However the speed of the resulting computer does not increase linearly with the number of procesors because of synchronization constraints. Hence it is essential to gain a theoretical understanding of synchronization constraints in order to improve the perfomance of these systems.

Computer networks of various sizes ranging from local area networks to intercontinental networks have proliferated since the early seventies. Most of the fundamental new problems that these networks posed were related to the distributed nature of the algorithms involved in controlling their operation [38]. This in turn lead to increased appreciation of the importance of synchronization constraints since they arise in most distributed algorithms.

Some well known examples of systems with synchronization constraints are given below.

(1) There are two general approaches in a communication network for transmitting messages, namely circuit switching and store-and-forward packet switching [15]. In the case of circuit switching, before a session between a source and destination node can be set up, it is necessary to acquire all the links that connect them together. Once the links are acquired, they are retained for the duration of

the session. Since all the messages are sent sequentially over the same path, they arrive at the destination in the same order that they were sent. In the store-and-forward approach, each session is initiated without reserving any links. Rather, one packet or message is transmitted on the links hop-by-hop until it reaches its destination. In some architectures, one or more paths are set up when a session is initiated and maintained for the duration of the session and all the packets are sent over those paths. Such multiple paths are useful for the following reasons. If one of the paths becomes faulty, then the others can take on its function, leading to a fault tolerant system. Multiple paths also help in distributing the traffic more evenly in the network. This scheme however has the disadvantage that packets may arrive at the destination node in a different order than the one in which they were sent due to variable network delays over different links. Hence there is some delay incurred due to the necessity of putting the packets back in order before they are presented to the destination computer. This is known as *resequencing delay*, and as explained below, it arises in a wide variety of situations, ranging from interconnection networks of a multi-processor computer to concurrency control schemes in a distributed data base.

(2) In most computer languages that support concurrent programming, there is a construct called a *fork* that transforms a sequential code into a number of concurrent parts which run simultaneously on different processors [13]. There is another construct called a *join* which reunites all these parts together into a sequential code. The synchronization constraint is that every process should communicate with every other process at the join before the execution can proceed . This is a very strong requirement in that if the running times of the parts are very different, this can lead to significant delays in the system which may neutralize some of the speed gained due to parallel operation.

(3) The fork-join operation is a coarse form of synchronization in a computer system. If we examine the system more closely, additional and more subtle forms of synchronization can be discerned which lead to a further degradation of perfomance. The processes created by the fork operation may need to exchange

variables among themselves to carry out the computation. If they share memory, they can read and write from it to carry out the communication. Although they can read from the common memory simultaneaously, they cannot write in it at the same time for this would lead to inconsistencies in the memory. Hence if more than one process attempts to access the memory, only one among them is given access, while the others have to wait their turn. Software constructs like monitors and semaphores enforce this synchronization constraint [13], [27]. If the concurrent algorithm is poorly designed, then the delay incurred may be large enough to make it much worse than the corresponding serial algorithm.

If the processes created by the fork operation are running on geographically distributed computers, as in a computer network, then they no longer have a common memory and the only way in which they can communicate with each other is then by sending messages across a communication link. This is the framework of a distributed programming methodology known as *Communicating Sequential Processes* [26]. If one of the processes needs an input variable from another process at some point in its execution, it waits at that point until the other process communicates that variable, thus incurring a synchronization delay.

(4) Another kind of synchronization constraint can be identified upon examining the system in examples (1) and (2) more closely, and arises due to the finiteness of the communication capacity of the system. Consider concurrent algorithms running on a multiprocessor machine with thousands of CPU and memory modules. Clearly a way needs to be found for efficiently connecting the CPU and memory. One such possible way is to connect each CPU with every memory module so that a CPU can instantaneously access any memory that it wishes. Such total connectivity leads to excessive wiring if the number of components is large. An alternative method consists in connecting the CPU's and memory together by means of what is known as an *interconnection* network [16]. This is usualy a multi-layered row of switches between the CPU and the memory. The use of switches leads to a significant reduction in the amount of wiring required. The switching strategy can be either circuit-switched or packet-switched. In the case

of a circuit-switched interconnection network, if a CPU module is accessing some memory module, then it occupies some of the switches in the interconnection network and some other communications from CPU to memory cannot take place. In the case of a packet-switched interconnection network, the kind of blocking described above does not take place but delays due to the queueing of the packets at the switch boxes may occur. One way to reduce this delay is to provide more than one path between each CPU and memory. As shown by Mitra [51], this can be done quite easily by adding additional layers of switches to the interconnection network. However due to multiple paths the message packets may not arrive at the memory in the same order that they left the CPU and this again translates into a resequencing delay.

(5) Data base control mechanisms in centralised and distributed systems display a rich variety of concurrent behaviour and synchronization constraints are thus of great importance in their performance.

Consider first a centralised multi-user data base. User programs that read or write data are known as transactions. Multiple transactions can read from a data base concurrently, whereas they cannot write into it at the same time. Data base consistency is expressed in terms of serializability. Concurrent execution of several transactions is correct if and only if its effect is the same as that obtained by running the same transactions serially in some order. The most popular method of enforcing serializability in the system is by means of *locks* [14], [76]. The database is partitioned into *items*, which are portions of the database that can be locked. There are two kinds of locks, read locks and write locks. A transaction wishing only to read an item, executes the readlock which prevents any other transaction from writing in that item while the readlock is in effect. However any number of transactions can hold a readlock on the item at the same time. A transaction wishing to change the value of an item first obtains a writelock on that item and no other transaction can either obtain a read or write lock on that item. A very simple rule which maintains consistency of the data base is known as the two phase protocol. It simply states that consistency is guaranteed if all read and write locks precede all unlocking steps in all transactions.

In a distributed data base, multiple copies of the items are kept in geographically distinct locations. The enforcement of consistency is more difficult because now we have to ensure that all the copies of the database are kept the same [14], [76]. There exist a number of algorithms that can enforce consistency, and we shall describe one such algorithm, known as the Le Lann ticketing scheme [45]. This algorithm provides a nice illustration of the resequencing constraint arising in a practical situation. The basic idea behind the algorithm is to predefine a total order among the update transactions on an item, and to process them according to this order on all the sites. Since the system is distributed, there is only a partial ordering of the time of updates being generated at different sites. However a total order can be obtained by putting all the sites on a virtual communication ring, and circulating tokens for each database item on this ring. Updates generated at a site are allocated sequentially increasing ticket numbers and attached to the token when it visits that site. Before being attached, the values of the ticket numbers from that site are incremented by the maximum of the values of the tickets which were already in the token in order to get a total order among all tickets. When the token reaches a site which has a copy of the data base, it delivers all its tickets to the database manager. The database manager makes updates on the data base in the order of the tickets attatched to the updates. If an update, say the n^{th} one were missing, then the $(n+1)^{rst}$ and higher updates cannot be processed even though they may be present at the site. This situation reveals the presence of the resequencing constraint in the algorithm.

To conclude this section, we now provide a generic classification of the synchronization constraints which covers all the cases described in examples (1) – (5).

- **The arrival-arrival synchronization**

Consider a system where multiple resources are represented by servers. If an arriving customer requires service from more than one resource and if it can receive service from them simultaneously, then we can represent this situation by simultaneous arrivals of customers to the resources triggered by the arrival of a single customer to the system.

- **The departure-departure synchronization**

  Consider the situation where several related processes are running simultaneously on different processors, and none of them are allowed to leave the system unless all the other processes to which they are related do so. This defines the departure-departure synchronization constraint.

- **The departure-arrival synchronization**

  This is the cause of the resequencing delay, which is the main topic of this thesis and can be considered to be a special case of the departure-departure kind of synchronization. If a system has more than one server, then the order in which the customers leave the system may not be the same as the order in which they entered it. The departure-arrival synchronization constrains the customers to leave the system in the same order as the one in which they entered. As a result of this constraint, a customer may have to suffer an additional delay.

We provide an example of a system which belongs to the class of systems with the departure-departure synchronization but not to the class of systems with departure-arrival synchronization. Consider a $G/G/K$ queue with bulk arrivals, with the constraint that no customer in a bulk can leave the system, unless all the customers in that bulk have completed service. Then clearly this system exhibits the departure-departure synchronization but not the departure-arrival synchronization, since bulks may not leave the system in the same order in which they entered it.

1.3 Mathematical Modelling of Systems with Synchronization Constraints

In this section we review the few mathematical models of systems with synchronization constraints for whom some analytical results are known. In Section 1.4 we give a more detailed account for systems with resequencing.

The modelling methodology borrows mainly from queueing theory, though some researchers have used a Petri-net representation for these systems. Queueing theory has been used to model resource sharing systems of which of systems with synchronization constraints are typical examples. Closed form solutions can be obtained for certain kinds of queueing networks, under assumptions that esentially imply a weak coupling between queues in the network [12], [36]. However, networks with synchronization constraints belong to a different class of systems due to the extremely strong coupling that exists throughout the network. An exact approach quickly results in an enormous increase in the size of the state space and makes the model both analytically and computationally intractable. At the present time, queueing theoretic methods do not handle such complexity very well, if at all and approximation techniques thus become important in analysing their behaviour.

- **Multiple Buffer Fork-Join queueing systems**

In a fork-join system, there is a finite number K of servers and a job upon arrival gets split up into K tasks with each part going to a queue served by a different server. After a task finishes service, it waits in an output buffer until the $K-1$ other tasks associated with it have finished service. So this system has arrival-arrival as well as departure-departure synchronization. When the system has two servers, the model with Poissonian arrivals was investigated for exponential service times by Flatto and Hahn [20], and for general service times by Baccelli [2]. These authors gave the generating function of the queue size distributions in equilibrium. In the so-called Markovian case, Flatto and Hahn applied the uniformization technique while in the more general case studied by Baccelli, the solution involved the transformation of the problem into a Riemann-Hilbert type boundary value problem. The sheer complexity involved in analyzing the case $K = 2$ makes it very unlikely that the more general case $K > 2$ will be analytically tractable.

Hence the focus shifted to finding at least approximate bounds for the performance measures in the general case. Seminal work in this area was done by Baccelli, Makowski and Shwartz [6], [7] who obtained computable upper and lower bounds for the system time of a customer in a fork-join queueing system with K servers. Here the system time is defined as the time between the arrival of a job into the system and its departure. The technique used by these authors was based on the theory of stochastic ordering. Recently, Baccelli, Massey and Towsley [8] have extended these bounding results to a more general acyclic fork-join network. In [75] tight bounds were developed for the case $K = 2$ under the Markovian assumption, by using matrix geometric techniques. In [54], tight bounds were developed for the general case of K queues by using a scaling approximation technique that was guided both by experimental and theoretical considerations.

- **Single buffer fork-join systems**

Towsley and Yu [75] considered a fork-join system with two servers in which arriving customers join a single queue, and get split into two tasks only when they reach the servers. Bounds were developed for this system by matrix geometric techniques and it was shown that the single queue has lower response times than a fork-join queue with $K = 2$. In [55] analysis was done for the case of K servers under the assumption that jobs are already split up into subtasks when they enter the common buffer (corresponding to bulk arrival). Bounds were developed for this system using stochastic ordering techniques. In [62], the single queue fork-join system was analysed for the case when the server operates with the processor sharing discipline.

- **Queues with locking**

These kind of queueing systems are very similar to the fork-join quueing systems and provide mathematical model of data-base concurrency mechanisms. There are K parallel servers and each arriving customer brings work to a subset of the K servers, instead to all the K servers as in the fork-join case. Mitra and Weinberger [52], [53] assumed that when an arriving customer finds one or more of the servers that it requests already blocked then it is lost. Under this assumption they found a product form solution for the equilibrium state probabilities. Surprisingly enough, if the assumption of blocking is removed, then exact analysis of this model becomes very difficult.

1.4 Systems with Resequencing

In this section, we review the resequencing models which have been analysed in the literature. As mentioned earlier, the resequencing problem arises whenever the order in which the customers leave the system is constrained to be the same as the order in which they enter the system.

- **Resequencing due to the $M/M/\infty$ queue**

This is the simplest resequencing model of its kind and the first one to be analyzed by Kamoun et al. [32]. In this model, there is a Poisson arrival stream of customers to an infinite server queue and the the service time in each one of the servers is exponentially distributed with the same parameter. After a customer finishes service, it leaves immediately if all the customers who arrived before it have finished their service. Otherwise it waits in a resequencing buffer until all the customers who arrived before it, finish their service. Kamoun et al. derived the steady state statistics of the total system time of a customer, the distribution of number of customers in the resequencing box and the statistics of the output process from the resequencing box. An interesting fact that emerges from the analysis of this model is the complicated nature of the output process which is a bulk departure process whose interdeparture times are correlated with each other and with the size of the bulks. This behaviour rules out exact analysis of models which consist of two resequencing systems in tandem, because the input process into the second system is in fact the output process from the first system, which as mentioned earlier is highly complex.

- **Resequencing due to the $M/G/\infty$ queue**

This model is similar in all respects to the model described above except for the fact that the service times are now identically distributed with some general distribution. This system was studied by Harrus and Plateau [23] who derived expressions for the above-mentioned perfomance measures Although the formulae become considerably more complicated due to the non-exponentiality of the service time distributions, the analysis is mostly straightforward and similar to the one carried out for the $M/M/\infty$ queue.

- **Resequencing due to a $M/M/\infty$ queue followed by a single server queue**

  This model was introduced to analyse the concurrency control scheme of Le Lann in a distributed data base [45]. It consists of a $M/M/\infty$ queue which disorders the input arrival stream, the customers are then put back in sequence and fed into a queue with exponential service times. As noted above, the output process from an $M/M/\infty$ queue with resequencing is a highly complicated bulk departure process with correlated bulk sizes and interdeparture times. This makes for a very difficult analysis of the single server queue with this kind of input process. Baccelli, Plateau and Gelenbe [5] circumvented this difficulty by focussing directly on the end-to-end delay and by using recursive sample path equations for the perfomance measures of interest. They derived an integral equation for the steady state distribution function of the end-to-end delay, by an analysis very similar to the classical analysis of the $GI/G/1$ queue. A solution to the integral equation gave closed form expressions for the distribution function.

- **Resequencing due to the $M/M/K/B$ system**

  In this resequencing system, K servers operate in parallel, each with a possibly different service rate, and the customers wait in a common buffer of size B. This model was analyzed by Yum and Ngai [80], who obtained the distribution of the waiting time in the resequencing buffer. Their final formula for the resequencing delay distribution depends upon the calculation of the buffer occupation probabilities in the $M/M/K/B$ queue with servers operating at different rates, and involves the solution of a numerical algorithm. Luke Lien [49] investigated the special case of an $M/M/2$ (i.e. $B = \infty$) queue with unequal service rates. To get a Markovian state representation he extended the state space of the $M/M/2$ queue in a very clever way and obtained the average resequencing delay. Unfortunately, Lien's technique does not seem to extend to more general situations such as $M/M/K$ queues.

- **Resequencing due to two $M/M/1$ queues in parallel**

  This resequencing system is composed of two $M/M/1$ queues in parallel with a Bernoulli switch routing customers to the two queues. By using sample path techniques, Jean-Marie [30] obtained the distribution of the resequencing delay. He also solved the problem of optimum static routing of customers to the the two queues so as to minimize their total system times.

1.4 The Main Results in the Thesis

The thesis is divided in two parts. Part I deals with some perfomance evaluation issues while Part II is devoted to the analysis of an optimal control problem.

Part (I) is subdivided in three chapters. Several structural properties of single stage resequencing systems are explored in Chapter 2. Though isolated results about some these systems have appeared in literature, only the simplest systems have been solved. We show how the perfomance measures of more complicated systems can be bounded by those of simple systems. We also investigate the variation of resequencing delay with the number of servers in a multi-server disordering system. In Chapter 3, we prove several interesting properties concerning multi-stage disordering systems with resequencing. Because of their complicated nature, there are no previous analytical results in the literature concerning these systems. We investigate the variation of end-to-end delay with resequencing strategies, in particular we show that end-to-end resequencing is superior to hop-by-hop resequencing when the disordering system has an infinite number of servers. We also show that several structural properties concerning single stage multiple server systems, carry over to multistage multi-server systems provided the customers are resequenced after each queue. In Chapter 4, we provide a detailed analysis of the $M/M/2/B$ queue with resequencing using matrix geometric techniques.

The model under investigation in Part II is a M/M/2 queue with heterogenous service rates and resequencing of customers. We deal with the problem of optimal dynamic allocation of customers to the servers so as to minimize their system times. There is no prior work in the literature concerning dynamic optimization in the presence of synchronization constraints. In Chapter 5 we identify the optimal policy for assigning customers to the faster server which states that it should be kept busy whenever possible. In Chapter 6 we prove that the optimal policy which assigns customers to the slower server is independent of the number of customers in the resequencing buffer and is of the threshold type in the number of customers in the main queue buffer.

CHAPTER II

STRUCTURAL PROPERTIES OF RESEQUENCING SYSTEMS

2.1 Introduction

Our objective in this chapter is to identify some structural properties of certain kinds of resequencing systems. Since most resequencing systems are analytically intractable, these structural properties help in obtaining bounds for them in some cases. The main inspiration for this kind of analysis is in [7] and [8] were bounds were developed for systems of fork join queues.

The chapter is organized as follows. In Section 2.2, a basic representation result is presented that gives the system delays in general resequencing systems in recursive sample path form. In Section 2.3, we develop some basic bounding methodologies which we shall apply in the remainder of the chapter to obtain bounds for different kinds of resequencing systems. These methodologies are developed without any particular model in mind. The models will be introduced with the corresponding notation as the need arises, later on in the chapter. In Sections 2.4 and 2.5, we identify bounds for infinite server queues and finite server queues with resequencing, repectively.

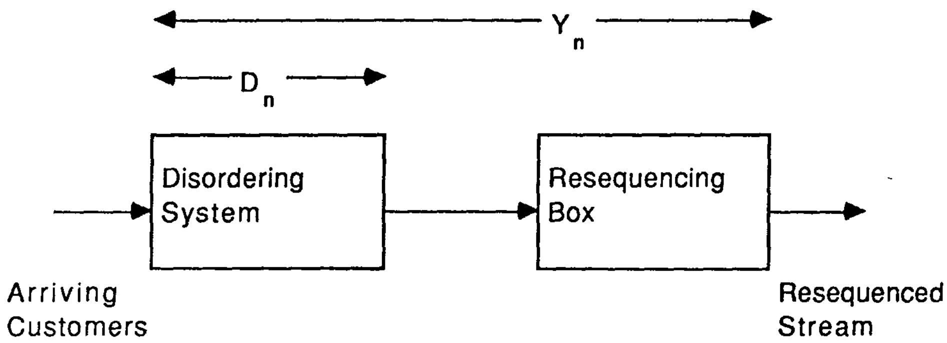

Fig 2.2.1 Basic Resequencing System

2.2 Basic Representations of Resequencing Systems

In this section we introduce a generic resequencing model, from which specific resequencing structures can be recovered as special cases. There is a stream of customers which enter a disordering system, and leave in an order different from the one in which they entered it (Fig 2.2.1). After leaving the disordering system, they wait in a resequencing buffer until all customers who entered the disordering system prior to them, have left it.

We now define some RV's that are useful in discussing properties of this system. Let the sequences of RV's $\{T_n\}_0^\infty$ and $\{D_n\}_0^\infty$ be defined on some probability space $\{\Omega, \mathbb{F}, P\}$. Here, T_n and D_n represent the time of arrival of the n^{th} customer into the system and its disordering delay respectively. We adopt the convention that the 0^{th} customer comes at time $t = 0$, so that $T_0 = 0$. In terms of these RV's define the following quantities for all $n = 0, 1 \ldots,$

d_n: Departure instant of the n^{th} customer from the system.

Y_n: End-to-end delay of the n^{th} customer (i.e., $Y_n = d_n - T_n$).

W_n: Waiting time of the n^{th} customer in the resequencing box (i.e., $W_n = Y_n - D_n$).

A_n: Interarrival interval between the $(n+1)^{rst}$ and the n^{th} customer (i.e., $A_n = T_{n+1} - T_n$).

Various kinds of disordering systems can be realized by assuming different statistical structures on the sequence $\{D_n\}_0^\infty$. For example, if the delay sequence $\{D_n\}_0^\infty$ is an i.i.d sequence which is independent of the interarrival sequence $\{A_n\}$, then the disordering system corresponds to an $GI/G/\infty$ queue. Similarly we can realize the disordering system as a $G/G/K$ queue or a system of K parallel $G/G/1$ queues by imposing a particular structure on $\{D_n\}_0^\infty$.

We now proceed to prove Theorem 2.2.1, which provides a recursive relationship between the sequences $\{Y_n\}_0^\infty$, $\{D_n\}_0^\infty$ and $\{A_n\}_0^\infty$ defined earlier.

Theorem 2.2.1. *Consider a resequencing system of the type shown in Fig 2.2.1. If there is no initial load on the system, the end-to-end delays $\{Y_n\}_0^\infty$ are given by the relations,*

$$Y_0 = D_0, \tag{2.1a}$$

and

$$Y_{n+1} = \max\{D_{n+1}, Y_n - A_{n+1}\} \qquad n = 0, 1, \ldots \tag{2.1b}$$

Proof. Since there is no initial load in the system by assumption, the first customer in the system will not undergo any resequencing delay and (2.1a) is therefore immediate.

In order to prove equation (2.1b), consider the $(n+1)^{rst}$ customer. His resequencing delay will be zero if the n^{th} customer has left the system at the time when he leaves the *disordering* subsystem i.e.,

$$Y_{n+1} = D_{n+1} \quad \text{if} \quad T_{n+1} + D_{n+1} > T_n + Y_n \qquad n = 0, 1 \ldots \tag{2.2}$$

If the n^{th} customer has not left the system at the time the $(n+1)^{rst}$ customer leaves the disordering subsystem, then the $(n+1)^{rst}$ customer will experience a resequencing delay of duration $T_n + Y_n - (T_{n+1} + D_{n+1})$, hence

$$Y_{n+1} = D_{n+1} + [T_n + Y_n - (T_{n+1} + D_{n+1})] \quad \text{if} \quad T_{n+1} + D_{n+1} < T_n + Y_n \tag{2.3}$$

By combining (2.2) and (2.3), it is plain that

$$Y_{n+1} = \max\{D_{n+1}, Y_n - (T_{n+1} - T_n)\} \qquad n = 0, 1 \ldots$$

since $A_{n+1} = T_{n+1} - T_n$, and this proves (2.1b). ∎

The recursion (2.1) was first derived by Baccelli, Gelenbe and Plateau [5] albeit in a different context since they were trying to estimate the end-to-end delay in an infinite server resequencing system followed by a single server queue. Equation (2.1) is very basic since it provides us with a relationship between the disordering delays and the end-to-end system delays. It will be used in a number of places in making stochastic comparisons in this chapter as well as the next one.

2.3 Some General Bounding Methodologies

Consider the sequences $\{A_n\}_1^\infty$ and $\{D_n\}_0^\infty$ of $\mathbb{R}_+$-valued RV's defined on some probability triple $(\Omega, \mathbb{F}, P)$. Assume these RV's to have finite means, i.e.,

$$E[A_n] < \infty, \qquad E[D_n] < \infty \qquad n = 1, 2, \ldots$$

For any sub σ-field $\mathbb{D}$ of $\mathbb{F}$, define the sequences $\{A_n(\mathbb{D})\}_0^\infty$ and $\{D(\mathbb{D})\}_0^\infty$ by

$$D_n(\mathbb{D}) = E[D_n \mid \mathbb{D}], \qquad A_n(\mathbb{D}) = E[A_n \mid \mathbb{D}] \qquad n = 0, 1 \ldots (2.4)$$

The sequences $\{Y_n\}_0^\infty$ and $\{Y(\mathbb{D})\}_0^\infty$ are then defined in terms of the above-mentioned sequences by the recursions,

$$Y_{n+1} = \max\{D_{n+1}, Y_n - A_{n+1}\} \qquad n = 0, 1 \ldots (2.5)$$

and

$$Y_{n+1}(\mathbb{D}) = \max\{D_{n+1}(\mathbb{D}), Y_n(\mathbb{D}) - A_{n+1}(\mathbb{D})\} \qquad n = 0, 1 \ldots (2.6)$$

with $Y_0 = D_0$ and $Y_0(\mathbb{D}) = D_0(\mathbb{D})$. The recursion (2.5) describes the evolution of system times of customers passing through a disordering system followed by resequencing, where A_n and D_n represent the interarrival time between the n^{th}

and the $(n-1)^{rst}$ customer, and the disordering delay for the n^{th} customer, respectively. A similar interpretation is available for (2.6) in terms of the sequences $\{A_n(\mathbb{D})\}_0^\infty$ and $\{D_n(\mathbb{D})\}_0^\infty$. We now state Theorem 2.3.1, which establishes an ordering property between the sequences $\{Y_n\}_0^\infty$ and $\{Y_n(\mathbb{D})\}_0^\infty$.

Theorem 2.3.1. *For any sub σ-field $\mathbb{D}$ of $\mathbb{F}$, the inequalities*

$$Y_n(\mathbb{D}) \leq E[Y_n \mid \mathbb{D}] \qquad n = 0, 1 \ldots \quad (2.7)$$

hold true where the sequences $\{Y_n\}_0^\infty$ and $\{Y_n(\mathbb{D})\}_0^\infty$ are defined as in (2.5)-(2.6).

Proof. We will provide an inductive proof of equation (2.7). For $n = 0$, it is clear that $Y_0 = D_0$ and $Y_0(\mathbb{D}) = D_0(\mathbb{D})$, so that

$$Y_0(\mathbb{D}) = D_0(\mathbb{D}) = E[D_0 \mid \mathbb{D}] = E[Y_0 \mid \mathbb{D}]$$

by invoking (2.4), i.e., (2.7) is satisfied for $n = 0$. As the induction step, assume that (2.7) is true for some $n = m \geq 1$, we will show that (2.7) also holds for $n = m + 1$. Applying Jensen's inequality to equation (2.5), we obtain

$$E(Y_{m+1} \mid \mathbb{D}) \geq \max\{E[D_{m+1} \mid \mathbb{D}], E[Y_m \mid \mathbb{D}] - E[A_{m+1} \mid \mathbb{D}]\} \qquad (2.8)$$

Using (2.4), we now get from the induction hypothesis that

$$\begin{aligned} E[Y_{m+1} \mid \mathbb{D}] &\geq \max\{D_m(\mathbb{D}), Y_m(\mathbb{D}) - A_{m+1}(\mathbb{D})\} \\ &= Y_{m+1}(\mathbb{D}) \end{aligned} \qquad (2.9)$$

Thus (2.7) holds for $n = m + 1$, and since it holds for $n = 0$, it holds by induction for all n. ∎

Theorem 2.3.1 holds for very general resequencing systems. Indeed the proof of Theorem 2.3.1 does not require any assumptions about the statistical nature of the sequences $\{A_n\}_0^\infty$ and $\{D_n\}_0^\infty$, which may be non-stationary and/or even correlated with each other. As in [7], we now provide an interpretation for (2.7) in terms of stochastic ordering. To that end, let f be any integrable convex non-decreasing function on $f : \mathbb{R}_+ \to \mathbb{R}$. From Jensen's inequality, it is clear that

$$E[f(Y_n) \mid \mathbb{D}] \geq f[E(Y_n \mid \mathbb{D})] \qquad n = 0, 1 \ldots \quad (2.10)$$

and using the increasing monotonocity of f, it follows from (2.7) that,

$$E[f(Y_n) \mid I\!D] \geq f[Y_n(I\!D)] \qquad n = 0, 1 \ldots (2.11)$$

Taking expectations on both sides of (2.11), we finally obtain the inequality

$$E[f(Y_n)] \geq E[f(Y_n(I\!D)] \qquad n = 0, 1 \ldots (2.12)$$

which reads as the defining relation for the convex-increasing stochastic ordering henceforth denoted as $\leq_{ci}$. Some useful properties of this ordering are stated in Appendix A.

We now give an interpretation of Theorem 2.3.1 in terms of the convex increasing ordering concept. Note that (2.4) implies that for each $n = 0, 1 \ldots$, $(A_n(I\!D), D_n(I\!D)) \leq_{ci} (A_n, D_n)$, since for any any convex increasing function f, Jensens inequality and the increasing nature of f imply that

$$E[f(D_n, A_n) \mid I\!D] \geq f([D_n(I\!D)], [A_n(I\!D)]) \tag{2.13}$$

so that

$$E[f(D_n, A_n)] \geq E[f(D_n(I\!D), A_n(I\!D)] \tag{2.14}$$

Also equation (2.7) can be written as $Y_n(I\!D) \leq_{ci} Y_n$. Hence the intuitive meaning of (2.7) is that greater variability in the inter-arrival times *and* the disordering delays, causes larger variability in the total system times. This is a very general statement especially because no assumptions are made about the inter-arrival or disordering sequences. By giving various structures to the disordering system, we can derive some special cases of Theorem 2.3.1. Examples are given later in Sections 2.4–2.5.

In the remainder of Section 2.3, we prove another result which sheds some light on the bounding problem for resequencing systems. Theorem 2.3.1 presented a result on convex orderings in resequencing systems, while Theorem 2.3.2 below deals with the strong stochastic ordering in resequencing systems. The advantage of using strong stochastic ordering is that proofs can be reformulated in terms of direct sample path comparisons between the systems of interest, which considerably reduces their complexity. Extensive use will be made of sample path comparisons in the remainder of the thesis. The reader may consult Appendix B for a review of some useful properties of strong stochastic orderings.

The system under consideration is the one described by (2.5), and again no assumption is made about the statistical properties of the sequences $\{A_n\}_0^\infty$ and $\{D_n\}_0^\infty$. Using the techniques in [59], we are able to prove that strong stochastic ordering between the disordering delays and inter-arrival times of two systems, implies stochastic ordering between their system times.

Consider a disordering system with resequencing and zero initial load. Let the first arrival occur at time zero and the nth arrival at T_{n+1}. Let $A_{n+1} = (T_{n+1} - T_n)$ for all $n = 1, 2 \ldots$. Order the sequences $\{A_n\}_1^\infty$ and $\{D_n\}_0^\infty$ as $(D_0, A_1, D_1, A_2, \ldots)$ and define the *transition functions* $\{p_n(.)\}_0^\infty$ as

$$\begin{aligned}
p_0(z) &= P[D_0 \leq z] \\
p_1(x; z) &= P[A_1 \leq z \mid D_0 = x] \\
p_2(x, y; z) &= P[D_1 \leq z \mid D_0 = x, A_1 = y] \\
&\vdots
\end{aligned} \tag{2.15}$$

The methodology of the proof is based on the *coupling* argument and closely follows the discussion in [59]. Given two resequencing systems with their transition functions satisfying certain inequalities, we construct two new queueing systems on a common probability space such that the new systems individually have the same probablistic structure as the original systems, and the system time of one lies entirely below the system time of the other.

Let $(\Omega, \mathbb{F}, P)$ be a fixed probability space on which is defined a sequence $\{\xi_n\}_0^\infty$ of independent RV's, each uniformly distributed on $(0, 1)$. The following Lemma is stated without proof. The reader may consult [60] for additional details.

Lemma 2.3.1. *Define the RV's* $\{A_n^*\}_1^\infty$ *and* $\{D_n^*\}_0^\infty$ *on* $(\Omega, \mathbb{F}, \mathbb{P})$ *by*

$$\begin{aligned} D_0^* &= \inf\{s \in [0,\infty) : p_0(s) \geq \xi_0\}, \\ A_1^* &= \inf\{t \in [0,\infty) : p_1(D_0^*; t) \geq \xi_1\} \\ D_1^* &= \inf\{s \in [0,\infty) : p_2(D_0^*, A_1^*; s) \geq \xi_2\} \end{aligned} \tag{2.16}$$

and so on. Then RV's $\{A_0, D_0, A_1, \ldots\}$ *and* $\{A_0^*, D_0^*, A_1^* \ldots\}$ *have the same finite dimensional distributions.*

Lemma 2.3.1 is a generalization of the *standard construction* of Lehmann [44] (see also Appendix B). We can now state the main result.

Theorem 2.3.2. *Suppose* $\{D_{01}, A_{11}, D_{11}, \ldots\}$ *and* $\{D_{02}, A_{12}, D_{12}, \ldots\}$ *are two resequencing systems, having transition functions* $\{p_n^1\}_0^\infty$ *and* $\{p_n^2\}_0^\infty$ *respectively. Assume that*

$$p_0^1(z) \leq p_0^2(z) \tag{2.17a}$$

and

$$(-1)^n p_n^1(x_0, \ldots, x_{n-1}; z) \leq (-1)^n p_n^2(y_0, \ldots, y_{n-1}; z) \qquad n = 1, 2, \ldots \tag{2.17b}$$

when $(-1)^j x_j \leq (-1)^j y_j, j = 0, 1 \ldots$ *Using the procedure of Lemma 2.3.1, construct two other resequencing systems on the space* $(\Omega, \mathbb{F}, P)$*, made up of the sequences* $\{A_{01}^*, D_{01}^*, A_{11}^*, \ldots\}$ *and* $\{A_{02}^*, D_{02}^*, A_{12}^*, \ldots\}$ *respectively. Under these conditions, the comparisons*

$$\begin{aligned} A_{n2}^* &\leq A_{n1}^*, \qquad n = 1, 2 \ldots \\ D_{n2}^* &\geq D_{n1}^*, \qquad n = 0, 1 \ldots \end{aligned} \tag{2.18a}$$

hold and consequently

$$Y_{n1}^* \leq Y_{n2}^*, \qquad n = 0, 1 \ldots \tag{2.18b}$$

which is equivalent to

$$Y_{n1} \leq_{st} Y_{n2} \qquad n = 0, 1 \ldots \tag{2.18c}$$

for the original two systems.

Proof. The proof of (2.18a) is straightforward and proceeds by fixing $\omega \in \Omega$ and applying Lemma 2.3.1.

Since the transition functions of D_{01} and D_{02} are unconditional, the classical proof by Lehmann [44], applies to yield

$$D^*_{02} \geq D^*_{01}. \tag{2.19}$$

For the general case, we will illustrate the technique by proving that $A^*_{12} \leq A^*_{11}$ and leave the proof of the other inequalities to the reader. By (2.16), it follows that

$$p^2_1(D^*_{02}; A^*_{12}) \geq \xi_1 \tag{2.20}$$

Using (2.17b) with $n = 1$, we get that

$$p^1_1(D^*_{01}; A^*_{12}) \geq \xi_1. \tag{2.21}$$

and by the definition of A^*_{11}, it now follows that

$$A^*_{12} \geq A^*_{11}. \tag{2.22}$$

We now give a proof for (2.18b) by induction. Recall that the sequences $\{Y^*_{n1}\}^\infty_0$ and $\{Y^*_{n2}\}^\infty_0$ are defined by the equations

$$\begin{aligned} Y^*_{(n+1)i} &= \max\{D^*_{(n+1)i}, Y^*_{ni} - A^*_{(n+1)i}\} \\ Y^*_{0i} &= D^*_{0i} \end{aligned} \qquad n = 0, 1 \ldots \tag{2.23}$$

with $i = 1, 2$. The case $n = 0$ for (2.18b) now follows easily from the definition in (2.23) and (2.18a). Assume that (2.18b) holds for $n = m \geq 0$, i.e.,

$$Y^*_{m1} \leq Y^*_{m2} \tag{2.24}$$

From (2.18a) and (2.24), it follows that

$$Y^*_{m1} - A^*_{(m+1)} \leq Y^*_{m2} - A^*_{(m+1)2}, \tag{2.25}$$

and combining (2.25) with (2.18a), we finally get

$$\max\{D^*_{(m+1)1}, Y^*_{m1} - A^*_{(m+1)1}\} \le \max\{D^*_{(m+1)2}, Y^*_{m2} - A^*_{(m+1)2}\} \tag{2.26}$$

hence (2.18b) holds for $n = m + 1$ and the induction step is now completed.

That (2.18b) implies (2.18c) is immediate since

$$[Y^*_{n2} \le x] \subset [Y^*_{n1} \le x] \qquad n = 0, 1 \ldots (2.27)$$

Thereforel

$$P[Y_{n2} \le x] \le P[Y_{n1} \le x] \qquad n = 0, 1 \ldots (2.28)$$

and the proof is completed. ∎

We now specialize Theorem 2.3.2 to the case when the interarrival times are the same for each queue and do not depend upon the preceding service times.

Corollary 2.3.1. *Suppose that*

$$p^1_{2n+1}(x_1, \ldots, x_{2n}; z) = p^2_{2n+1}(x_1, \ldots, x_{2n}; z) \qquad n = 0, 1 \ldots (2.28a)$$

and both are independent of $x_2, x_4, \ldots, x_{2n}$. *Suppose also*

$$p^1_{2n}(x_0, \ldots, x_{2n-1}; z) \le p^2_{2n}(x_1, y_2, x_3, y_4, \ldots, x_{2n-1}; z) \qquad n = 0, 1 \ldots (2.28b)$$

whenever $y_{2j} \ge x_{2j}, j = 1, 2, \ldots, n-1$. *Then*

$$Y_{n1} \le_{st} Y_{n2} \qquad n = 0, 1 \ldots (2.29)$$

∎

Corollary 2.3.1 is a trivial consequence of Theorem 2.3.2, and we omit its proof. In most of the cases in the sequel, when we have to show that two resequencing systems with the same arrival process are strongly stochastically ordered, we try to show that in some sample space, the disordering delays of one of them lies entirely below the disordering delay of the other. This implies, by the proof of Theorem 2.3.2, that the system delays of the original two systems are strongly ordered.

2.4 The Infinite Server Case

This section contains the most elementary applications of Theorem 2.3.1. We first show that for $G/G/\infty$ systems, deterministic inter-arrival and disordering sequences achieve the lower bound for the end-to-end delay, among all possible distributions for these sequences. This turns out to be a direct consequence of Theorem 2.3.1. Next, taking advantage of the special structure of the $GI/G/\infty$ queue, we strengthen the result of Theorem 2.3.1, and obtain a computable upper bound for the end-to-end delay.

We will make the following assumption.

(A1) The RV's $\{A_n\}_1^\infty$ and $\{D_n\}_0^\infty$ form independent sequences of RV's.

Assumption (A1), defines a $G/G/\infty$ queue with resequencing. The RV's $\{D_n\}_0^\infty$ and $\{A_n\}_0^\infty$ are constructed on some probability space $(\Omega, \mathbb{F}, \mathbb{P})$ and let $\mathbb{D}$ be some sub σ-field of $\mathbb{F}$. Construct another $G/G/\infty$ queue in which the disordering and the interarrival sequences are given by $\{D_n(\mathbb{D})\}_0^\infty$ and $\{A_n(\mathbb{D})\}_0^\infty$ respectively where

$$D_n(\mathbb{D}) = E[D_n \mid \mathbb{D}] \quad \text{and} \quad A_n(\mathbb{D}) = E[A_n \mid \mathbb{D}]. \qquad n = 0, 1 \ldots (2.30)$$

If the sequences $\{Y_n\}_0^\infty$ and $\{Y_n(\mathbb{D})\}_0^\infty$ are defined by (2.5)-(2.6), then Theorem 3.3.1 immediately tells us that

$$Y_n(\mathbb{D}) \le E[Y_n \mid \mathbb{D}] \qquad n = 0, 1 \ldots (2.31)$$

Under the assumptions (A1) and (A2), where (A2) is

(A2) The σ-field $\mathbb{D}$ is generated by the sequence $\{D_n\}_0^\infty$.

we obtain,

$$E[D_n \mid \mathbb{D}] = D_n \qquad \text{and} \qquad E[A_n \mid \mathbb{D}] = E[A_n] \qquad n = 0, 1 \ldots (2.32)$$

From equations (2.6) and (2.32), we now get

$$Y_{n+1}(\mathbb{D}) = \max\{D_{n+1}, Y_n(\mathbb{D}) - E[A_{n+1}]\} \qquad n = 0, 1 \ldots (2.33)$$

But the recursion (2.33) corresponds to a $D/G/\infty$ system, with the same disordering delays as the original system but with deterministic interarrival times whose constant value equals the mean of the interarrival times of the original system. The system times for each customer in this queue is smaller than his system time in the original queue, in the convex increasing sense.

This observation can be used to derive a computable upper bound for the $D/M/\infty$ queue with resequencing, as shown next. Exact analysis of the $D/M/\infty$ queue with resequencing is difficult because of the complicated nature of the buffer occupation probability formulae for the $D/M/\infty$ queue [73]. However from the above discussion, it follows that the system times in a $D/M/\infty$ queue with resequencing are bounded from above in the convex increasing sense, by the system times in a $M/M/\infty$ queue with resequencing, a quantity which can be derived easily. Note that (2.33) establishes the ordering only for the transient case, but it can be carried over to the steady state case by noting that $\{Y_n\}_0^\infty$ converges in distribution to some random variable Y_∞ [72]. The convergence proof is essentially the one given in [5]. Hence we can write

$$Y_\infty(I\!D) \leq_{ci} Y_\infty \tag{2.34}$$

An expression for $E[Y]$ for the $M/M/\infty$ queue with resequencing is now derived. Though the derivation is well known [28], [32], [23]; we nevertheless include it here for completeness.

Consider a $M/M/\infty$ queue with arrival rate λ and service rate μ. The steady state probablitity that there are n customers in the queue is given by

$$\Pr(n) = \frac{\rho^n}{n!}\exp^{-\rho} \qquad n = 0, 1 \ldots \tag{2.35}$$

where $\rho = \lambda/\mu$. Suppose a customer arrives into the system and finds n customers in the process of being served. Because of the resequencing constraint, the tagged customer cannot leave the system unless all those n customers have also exited from the system (including the resequencing box). Hence, the system time

Y_∞ of the tagged customer is distributed according to the maximum of $(n+1)$ exponentially distributed RV's and routine standard computations yield

$$E[Y_\infty \mid n] = E[Y_\infty \mid \text{tagged customer finds n customers}]$$
$$= \sum_{i=1}^{n+1} \frac{1}{n\mu} \tag{2.36}$$

Since the distribution of the number of customers at the arrival instants coincides with the stationary distribution in the queue by the Poissonian nature of the arrivals, (by removing the conditioning in (2.36) and using (2.35)), we get

$$\begin{aligned} E[Y_\infty] &= \sum_{n=0}^{\infty} E[Y_\infty \mid n] P(n) \\ &= \sum_{n=0}^{\infty} \frac{\exp(-\rho)\rho^n}{n!} \sum_{i=1}^{n+1} \frac{1}{i\mu} \end{aligned} \tag{2.37}$$

Using (2.34) and (2.37), we can write

$$E[Y_\infty(\mathbb{D})] \leq E[Y_\infty] \tag{2.38}$$

where $E[Y_\infty]$ is given by (2.37) i.e. we have obtained a computable upper bound to the system time $E[Y(\mathbb{D})]$ of a $D/M/\infty$ queue with resequencing.

We now strengthen the result of Theorem 2.3.1 for $GI/G/\infty$ systems, in order to derive a computable upper bound for them.

Lemma 2.4.1. *Consider two $GI/G/\infty$ systems with resequencing, and no initial load with inter-arrival times $\{A_n^i\}_0^\infty$ and disordering delays $\{D_n^i\}_0^\infty$ and let $\{Y_n^i\}_0^\infty$ denote the system delay, $i = 1, 2$. If*

$$E[A_n^1] = E[A_n^2] \qquad n = 0, 1 \ldots \tag{2.39a}$$

and

$$A_n^1 \geq_{ci} A_n^2, \qquad D_n^1 \geq_{ci} D_n^2 \qquad n = 0, 1 \ldots \tag{2.39b}$$

then

$$Y_n^1 \geq_{ci} Y_n^2 \qquad n = 0, 1 \ldots \quad (2.40)$$

Proof. The proof proceeds by induction. It is plain that

$$Y_0^1 = D_0^1, \qquad Y_0^2 = D_0^2 \quad (2.41)$$

and (2.40) thus follows for $n = 0$ by (2.39b). Assume that (2.40) holds for some $n = m \geq 0$, i.e.,

$$Y_m^1 \geq_{ci} Y_m^2 \quad (2.42)$$

Since

$$D_{m+1}^1 \geq_{ci} D_{m+1}^2 \quad (2.43)$$

by assumption, and

$$-A_{m+1}^1 \geq_{ci} -A_{m+1}^2 \quad (2.44)$$

by (2.39a-b) and equation (A4) in Appendix A. Finally, since the sequences $\{A_n^i\}_1^\infty$ and $\{D_n^i\}_o^\infty$ are independent for $i = 1, 2$ and the function max is convex increasing, by Property 4 of Appendix A, it follows that

$$\max\{D_{m+1}^1, Y_m^1 - A_{m+1}^1\} \geq_{ci} \max\{D_{m+1}^2, Y_m^2 - A_{m+1}^2\} \quad (2.45)$$

and (2.40) also holds for $n = m + 1$, thus completing the induction.

∎

Note that if the sequences $\{Y_n^1\}_0^\infty$ and $\{Y_n^2\}_0^\infty$ converge in distribution to Y_∞^1 and Y_∞^2 then (2.40) also holds in the limit, i.e.,

$$Y_\infty^1 \geq_{ci} Y_\infty^2. \quad (2.46)$$

We will now use Lemma 2.4.1, to derive a computable upper bound for the $GI/G/\infty$ queue with resequencing. Consider a $GI/G/\infty$ queue with inter-arrival intervals and disordering sequences given by $\{A_n\}_0^\infty$ and $\{D_n\}_0^\infty$, having distributions F and G, respectively. The sequence $\{Y_n\}_0^\infty$ converges in distribution to the RV variable Y_∞. If the inter-arrival distribution F is NBUE (see Property 4 of ci orderings in Appendix A), with mean $\frac{1}{\lambda}$, then

$$F \leq_{ci} \exp(\frac{1}{\lambda}) \tag{2.47}$$

where $\exp(\frac{1}{\lambda})$ is the exponential distribution with mean $\frac{1}{\lambda}$. From Lemma 2.4.1 it is now easy to see that

$$Y_\infty \leq_{ci} Y'_\infty \tag{2.48}$$

where Y'_∞ is the system time in equilibrium of a $M/G/\infty$ queue with resequencing, fed by a Poissonion arrival stream of rate λ. An exact solution for $E[Y'_\infty]$ has been given in [23] in the form

$$E(Y'_\infty) = E(D) + \int_0^\infty (1 - \exp^{(-\lambda \int_t^\infty (1-F(u))\, du)})\, F(t)\, dt \tag{2.49}$$

and the mean of Y_∞ satisfies the bound

$$E[Y_\infty] \leq E[D] + \int_0^\infty (1 - \exp^{(-\lambda \int_t^\infty (1-F(u))\, du)}\, F(t)\, dt \tag{2.50}$$

2.5 The Finite Server Case

A number of structural properties of finite server resequencing systems are identified in this section, which is divided into four subsections. In Subsection 2.5.1, *ci*-orderings are used to identify structural properties of K parallel $G/G/1$ queues with resequencing, and in subsections 2.5–2.4, sample path comparison techniques are used to identify structural properties of $G/G/K$ queues with resequencing.

2.5.1 Determinism minimises response time

Let us assume the disordering system to be a set of K parallel $G/G/1$ queues with FCFS service discipline and a single input. We will use Theorem 2.3.1 to show that deterministic inter-arrival and service sequences achieve the lower bound for the end-to-end delay among all possible distributions for these sequences. The discussion in this section owes much to the treatment of the fork-join queue in [7].

We now introduce some additional notation. Let $(\Omega, I\!F, P)$ be a probability space on which several sequences of RV's are defined. The RV's $\{\sigma_n\}_0^\infty$ and $\{u_n\}_0^\infty$ are $I\!R_+^K$-valued RV's and $\{A_{n+1}\}_0^\infty$ are R_+ valued RV's. Here A_n is interpreted as the interarrival time between the $(n+1)^{rst}$ and the n^{th} customers. These RV's can be used to define a system of K parallel $G/G/1$ queues if the sequence $\{u_n^k\}_0^\infty, (1 \leq k \leq K)$ is chosen to be such that $\sum_{k=1}^K u_n^k = 1$, and

$$u_n^k = \begin{cases} 1, & \text{for some } k \in (1, \ldots, K); \\ 0, & \text{otherwise.} \end{cases} \qquad n = 0, 1 \ldots$$

In this case $\sigma_n.u_n^k$ can be interpreted as the effective service time of the n^{th} customer.

The RV's $\{W_n^k\}_0^\infty, 1 \leq k \leq K$, and $\{R_n\}_0^\infty$ are now defined recursively by

$$W_{n+1}^k = [W_n^k + u_n^k \cdot \sigma_n^k - A_{n+1}]^+, \qquad 1 \leq k \leq K, \qquad n = 0, 1 \ldots \quad (2.51)$$

with $W_0^k = 0$ for all $1 \leq k \leq K$ and

$$R_n = \sum_{k=1}^K u_n^k \cdot (W_n^k + \sigma_n^k) \qquad n = 0, 1 \ldots \quad (2.52)$$

Here, if $u_n^k = 1$, then W_n^k represents the waiting time of the n^{th} customer in the buffer of the k^{th} queue whereas R_n is interpreted as the system time of the n^{th} customer in the set of K parallel queues.

We will make the following assumptions :

(A3) The RV's $\{\sigma_n^k\}_0^\infty$, $\{u_n^k\}_0^\infty, 1 \le k \le K$ and $\{A_n\}_1^\infty$ have finite means.

(A4) There exists a sub-sigma field $\mathbb{D}$ of $\mathbb{F}$ with the property that for each $n = 0, 1 \ldots$, the RV u_n is conditionally independent of the σ-field $\mathbb{H}_n$ given $\mathbb{D}$ with

$$\mathbb{H}_n = \sigma\{\sigma_n\} \vee \sigma\{\sigma_m, u_m, A_{m+1}, 0 \le m < n\} \qquad n = 0, 1 \ldots$$

Next we define the R_+^K-valued RV's $\{W_n(\mathbb{D})\}_0^\infty$ componentwise by

$$W_{n+1}^k(\mathbb{D}) = [W_n^k(\mathbb{D}) + u_n^{k,\mathbb{D}} \cdot \sigma_n^{k,\mathbb{D}} - A_{n+1}^{\mathbb{D}}]^+, \qquad n = 0, 1, \ldots \text{(2.53)}$$

for all $1 \le k \le K$, and

$$R_n(\mathbb{D}) = \sum_{k=1}^{K} (W_{n+1}^k(\mathbb{D}) + \sigma_n^{k,\mathbb{D}}) \cdot u_n^{k,\mathbb{D}} \qquad n = 0, 1 \ldots \text{(2.54)}$$

where

$$\begin{aligned} u_n^{k,\mathbb{D}} &= E[u_n^k \mid \mathbb{D}] \\ \sigma_n^{k,\mathbb{D}} &= E[\sigma_n^k \mid \mathbb{D}] \\ A_{n+1}^{\mathbb{D}} &= E[A_{n+1} \mid \mathbb{D}] \end{aligned} \qquad n = 0, 1 \ldots \text{(2.55)}$$

for all $1 \le k \le K$ and for any σ field $\mathbb{D}$.

Theorem 2.5.1. *Let the RV W_0 be $\mathbb{D}$-measurable. Under the enforced assumptions (A3) and (A4), the inequalities*

$$W_n^k(\mathbb{D}) \le E[W_n^k \mid \mathbb{D}], \quad 1 \le k \le K \qquad n = 0, 1 \ldots \text{(2.56a)}$$

and

$$R_n(\mathbb{D}) \le E[R_n \mid \mathbb{D}] \qquad n = 0, 1 \ldots \text{(2.56b)}$$

hold true.

Proof. The RV W_n is $\mathbb{I}_n$-measurable for all $n = 0, 1 \ldots$. Hence by assumption (A4), the RV u_n is conditionally independent of the RV's $\{\sigma_n, W_n\}$ given the σ-field $\mathbb{D}$, and for all $1 \leq k \leq K$,

$$E[u_n^k \cdot W_n^k \mid \mathbb{D}] = u_n^{k,\mathbb{D}} \cdot W_n^{k,\mathbb{D}} \qquad n = 0, 1 \ldots (2.57a)$$

and

$$E[u_n^k \cdot \sigma_n^k \mid \mathbb{D}] = u_n^{k,\mathbb{D}} . \sigma_n^{k,\mathbb{D}} \qquad n = 0, 1 \ldots (2.57b)$$

We will prove inequalities (2.56a-b) by induction. Let (2.56a) hold for $n = m \geq 0$ so that

$$W_m^k(\mathbb{D}) \leq E[W_m^k \mid \mathbb{D}] \qquad 1 \leq k \leq K \tag{2.58}$$

Applying Jensen's inequality to (2.51) and using (2.57), we obtain

$$E[W_{m+1}^k \mid \mathbb{D}] \geq \left[E[W_m^k \mid \mathbb{D}] + u_n^{k,\mathbb{D}} \cdot \sigma_n^{k,\mathbb{D}} - A_{n+1}^{\mathbb{D}}\right]^+, \quad 1 \leq k \leq K \tag{2.59}$$

and the induction step (2.58) now yields

$$\begin{aligned} E[W_{m+1}^k \mid \mathbb{D}] &\geq \left[W_m^k(\mathbb{D}) + u_n^{k,\mathbb{D}} . \sigma_n^{k,\mathbb{D}} - A_{n+1}^{\mathbb{D}}\right]^+ \\ &= W_{m+1}^k(\mathbb{D}) \end{aligned} \tag{2.60}$$

where the last step follows from (2.53). Consequently, (2.58) holds for $n = m + 1$ and since by assumption the initial value W_0 is $\mathbb{D}$-measurable, the inequality holds for all $n = 0, 1 \ldots$.

We now provide a proof for (2.56b). By definition,

$$R_m = \sum_{k=1}^{K} u_m^k \cdot (W_m^k + \sigma_m^k), \tag{2.61}$$

so that

$$E[R_m \mid \mathbb{D}] = \sum_{k=1}^{K} u_m^{k,\mathbb{D}} \cdot \left[E[W_m^k \mid \mathbb{D}] + \sigma_m^{k,\mathbb{D}}\right] \tag{2.62}$$

by making use of (2.57). From (2.56a), we now obtain

$$E[R_m \mid \mathbb{D}] \geq \sum_{k=1}^{K} u_m^{k,\mathbb{D}} \cdot (W_m^k(\mathbb{D}) + \sigma_m^{k,\mathbb{D}}) \tag{2.63}$$
$$= R_m(\mathbb{D}).$$

∎

We will now use Theorems 2.3.1 and 2.5.1 to prove that determinism in interarrival or service times, minimises the system time of K parallel $G/G/1$ queues with resequencing.

Identify the RV variable R_n of Theorem 2.5.1 as the RV variable D_n in Theorem 2.3.1. The conditions of Theorem 2.3.1 are satisfied by the system of Theorem 2.5.1, and consequently if Y_n is the end-to-end delay inclusive of resequencing after a system of K parallel $G/G/1$ queues, then $Y_n(\mathbb{D}) \leq E[Y_n \mid \mathbb{D}]$ for all $n = 0, 1 \ldots$. Now, by making appropriate choices of $\mathbb{D}$ we can obtain lower bounds for the system.

(A) Consider the sub σ-field $\mathbb{D}_1$ of $\mathbb{F}$ given by

$$\mathbb{D}_1 = \sigma\{A_{n+1}, u_n, n = 0, 1 \ldots\}$$

under the conditions (A4) and (A5), where

(A5) $\mathbb{D}_1$ is independent of the σ-field $\sigma\{\sigma_n, n = 0, 1 \ldots\}$.

Conditions (A4) and (A5) will be satisfied if u_n is independent of the RV's $\{\sigma_m, 0 \leq m \leq n\}$ for all $n = 0, 1 \ldots$. Under these assumptions it is clear that for all $1 \leq k \leq K$,

$$u_n^{k,\mathbb{D}_1} = u_n^k$$
$$\sigma_n^{k,\mathbb{D}_1} = E[\sigma_n] \qquad n = 0, 1 \ldots \tag{2.64}$$
$$A_{n+1}^{\mathbb{D}_1} = A_{n+1}$$

Morever, (2.53) and (2.54) simplify to

$$W_{n+1}^k(\mathbb{D}_1) = \left[W_n^k(\mathbb{D}_1) + u_n^k \cdot E[\sigma_n] - A_{n+1}\right] \qquad n = 0, 1 \ldots \tag{2.65}$$

for all $1 \leq k \leq K$, and

$$R_n(\mathbb{D}_1) = \sum_{k=1}^{K} u_n^k \cdot [W_{n+1}^k(\mathbb{D}_1) + E[\sigma_n]] \qquad n = 0, 1 \ldots (2.66)$$

It follows the RV $R_n(\mathbb{D}_1)$ is the system time of the n^{th} customer in a $G/D/K$ which has the same arrival process as the original queue but with deterministic service times. By Theorem 2.4.1, the system with deterministic service times is a lower bound to our original system, with regard to the total system delay in the sense of convex increasing stochastic ordering.

(B) Consider the sub σ-field $\mathbb{D}_2$ given by where

$$\mathbb{D}_2 = \sigma\{\sigma_n, u_n, n = 0, 1 \ldots\}$$

under the assumptions (A4) and (A6), where

(A6) The σ-field $\mathbb{D}_2$ is independent of the σ-field $\sigma\{A_{n+1}, n = 0, 1 \ldots\}$

Conditions (A4) and (A6) will be satisfied if the RV u_n is independent of the RV's $\{A_{m+1}, 0 \leq m \leq n\}$ for all $n = 0, 1, \ldots$. From these assumptions, it follows that for all $1 \leq k \leq K$,

$$\begin{aligned} u_n^{k,\mathbb{D}_2} &= u_n^k \\ \sigma_n^{k,\mathbb{D}_2} &= \sigma_n^k \qquad n = 0, 1 \ldots (2.67) \\ A_{n+1}^{\mathbb{D}_2} &= E[A_{n+1}] \end{aligned}$$

By an argument analagous to that given in example (A), it follows that a system with deterministic interarrival times is a lower bound to our original system.

From this discussion, it follows that for the case $K = 2$, the end-to-end delays of two parallel $M/D/1$ queues or $D/M/1$ queues with resequencing are upper bounded by the end-to-end delays of two parallel $M/M/1$ queue with resequencing. If these end-to-end delays converge in distribution, their equilibrium values are ordered in the same way as their transient values. Formulae for the resequencing delay of two parallel $M/M/1$ with resequencing and Bernoulli loading, were obtained by Jean-Marie [30] and constitute a computable bound. These formulae are reproduced below.

Consider two parallel $M/M/1$ queues with resequencing. Let λ be the rate of the Poisson arrival process into the system, let p be the probability that the arriving customer joins the first queue and q the probability that he joins the second queue, and let the service rate be μ for both queues. Jean-Marie showed that for this system, the average resequencing delay is given by

$$E(W_\infty) = p\lambda w(\mu - p\lambda, \mu - q\lambda, q\lambda) + q\lambda w(\mu - q\lambda, \mu - p\lambda, p\lambda) \tag{2.68}$$

where

$$w(x, y, z) = \frac{xz}{y(x+y)(z+y)} \tag{2.69}$$

2.5.2 A lower bound for multi-server queues.

In what follows we will assume that each server in a multiserver queueing system has the same service distribution.

We now establish a result to the effect that finite server queues with resequencing, specifically the $GI/G/K$ queue or K parallel $GI/G/1$ queues, have system times that are lower bounded by the system times of a $GI/G/\infty$ queue with resequencing. This $GI/G/\infty$ queue has the same arrival process as the original system, and each of its servers has the same service distribution as those in the original system. The proof is based on a coupling argument, whereby we construct two new queueing systems on a common probability space such that the new systems individually have the same probabilistic structure as the original systems and for each sample path, the end-to-end delay of one system lies entirely below the end-to-end delay of the other system.

We will prove the result when the disordering system is a $GI/G/K$ queue. A similar proof applies when the disordering system is made of K parallel $GI/G/1$ queues, and we leave the details to the interested reader.

We now describe the recursion equations for the waiting times in a $GI/G/K$ queue. Let R^+ be a function which reorders the elements of a K dimensional vector in ascending order and replaces negative elements by zeros. For all $n = 0, 1 \ldots$, let $W_n = (W_{n1}, \ldots, W_{nK})$ be the vector of ascendingly ordered times remaining, measured from the time of the n^{th} arrival, until each of the various servers would first be available to serve the n^{th} customer. Keifer and Wolfowitz [31] derived the recursive relationship

$$W_{n+1} = R^+(W_n + \sigma_n e_1 - A_{n+1}1) \tag{2.70}$$

where e_1 and 1 are vectors given by

$$e_1 = (1, 0, \ldots, 0) \quad \text{and} \quad 1 = (1, \ldots, 1)$$

The waiting time of the n^{th} customer is given by the minimum element W_{n1} of the vector W_n. Due to the appearance of the minimum operator, convex ordering relationships of the type proved in Theorem 2.5.1 are not possible for $GI/G/K$ queues. The total disordering delay for the $(n+1)^{rst}$ customer, is given by the first element of the vector D_{n+1}, where

$$D_{n+1} = W_{n+1} + \sigma_{n+1}\mathbf{1} \qquad n = 0, 1 \ldots (2.71)$$

Denote the total system delay (i.e. end-to-end delay), for the $(n+1)^{rst}$ customer as Y_{n+1}, where

$$Y_{n+1} = \max\{D_{(n+1)1}, Y_n - A_{n+1}\} \qquad n = 0, 1 \ldots (2.72)$$

Now consider a $GI/G/\infty$ queue with inter-arrival and service times given by the sequences $\{\overline{A}_n\}_0^\infty$ and $\{\overline{D}_n\}_0^\infty$, so that the system time of the $(n+1)^{rst}$ customer $\overline{Y}_{n+1}$ is given by

$$\overline{Y}_{n+1} = \max\{\overline{D}_{n+1}, \overline{Y}_n - \overline{A}_{n+1}\} \qquad n = 0, 1 \ldots (2.73)$$

Assume that

(A7) The following relations hold

$$\begin{aligned} A_n &=_{st} \overline{A}_n \\ \sigma_n &=_{st} \overline{D}_n \end{aligned} \qquad n = 0, 1 \ldots$$

We now state the main result in this subsection.

Theorem 2.5.2. *The total system time of a customer in a $GI/G/K$ queue with resequencing is stochastically larger than his total system time in the $GI/G/\infty$ queue with resequencing, i.e.,*

$$Y_n \geq_{st} \overline{Y}_n \qquad n = 0, 1 \ldots (2.74)$$

under the assumption that both systems have an identical arrival process and service distributions.

Proof. On a fixed probability space $(\Omega, \mathbb{F}, P)$, define random variables $\{A_n\}_0^\infty$, $\{\overline{A}_n\}_0^\infty$, $\{\sigma_n\}_0^\infty$ and $\{\overline{D}_n\}_0^\infty$ such that

$$\begin{aligned} A_n &= \overline{A}_n \\ \sigma_n &= \overline{D}_n \end{aligned} \qquad n = 0, 1 \ldots (2.75)$$

This is always possible due to assumption (A7).

Since

$$W_{n+1} = R^+(W_n + \sigma_n e_1 - A_{n+1}1)$$
$$D_{n+1} = W_{n+1} + \sigma_{n+1}1 \qquad n = 0, 1 \ldots (2.76)$$

observe from (2.75)-(2.76) that

$$D_{n1} = W_{n1} + \sigma_n \geq \sigma_n = \overline{D}_n \qquad n = 0, 1 \ldots (2.77)$$

Using (2.77) and (2.72)-(2.73), we can recursively as in Theorem 2.3.2 that

$$Y_n \geq \overline{Y}_n \qquad n = 0, 1 \ldots (2.78)$$

The conclusion (2.74) now follows directly from (2.78).

∎

As mentioned earlier a similar result holds for a disordering system made of K $GI/G/1$ queues in parallel, and is stated below without proof.

Theorem 2.5.3. *The total system time of a customer in a system of K parallel $GI/G/1$ with resequencing, is stochastically larger than his total system time in a $GI/G/\infty$ queue with resequencing, provided both systems have identical inter-arrival and service distributions.*

∎

Theorems 2.5.2-3 are useful in obtaining computable lower bounds for finite server resequencing systems with a Poissonian arrival process, since in this case the lower bounds are given by the $M/G/\infty$ queue with resequencing, exact formulae for which are available in [23].

Theorems 2.5.2-3 in combination with Corollary 4.3.2 can also be used to obtain a lower bound for the total system time of a customer in a hop-by-hop resequencing system consisting of N stages, each of which is a multi-server queue in the following manner. We first replace all the multi-server queues with infinite server queues, and then replace the hop-by-hop resequencing system by an end-to-end resequencing system. It is clear that this will be a lower bound to the original system. If the arrival process into the system is Poissonian, then this lower bound is also computable.

2.5.3 An upper bound for the $M/G/K$ queue with resequencing.

The next result establishes an ordering relationship between the system times of two $GI/G/K$ queues with resequencing which have the same arrival process, and the service time of one of them is stochastically larger than the service time of the other. Let $\{\sigma_n^1\}_0^\infty$ and $\{A_n^1\}_0^\infty$ be the service and inter-arrival processes for the first queue and denote by $\{\sigma_n^2\}_0^\infty$ and $\{A_n^2\}_0^\infty$ be the corresponding quantities for the second queue. The delays in the two queues are given by the equations

$$\begin{aligned} D_{n+1}^1 &= W_{(n+1)1}^1 + \sigma_{n+1}^1 \\ D_{n+1}^2 &= W_{(n+1)1}^2 + \sigma_{n+1}^2 \end{aligned} \qquad n = 0, 1 \ldots (2.79)$$

with $D_0^1 = \sigma_0^1$ and $D_0^2 = \sigma_0^2$. Here the vectors W_{n+1}^1 and W_{n+1}^2 are given by

$$\begin{aligned} W_{n+1}^1 &= R^+(W_n^1 + \sigma_n^1 e - A_{n+1}^1 1) \\ W_{(n+1}^2 &= R^+(W_n^2 + \sigma_n^2 e - A_{n+1}^2 1) \end{aligned} \qquad n = 0, 1 \ldots (2.80)$$

where $W_0^1 = W_0^2 = 0$. The total system delays, including the resequencing delays, are then given by

$$\begin{aligned} Y_{n+1}^1 &= \max(D_{n+1}^1, Y_n^1 - A_{n+1}^1) \\ Y_{n+1}^2 &= \max(D_{n+1}^2, Y_n^2 - A_{n+1}^2) \end{aligned} \qquad n = 0, 1 \ldots (2.81)$$

with $Y_0^1 = D_0^1$ and $Y_0^2 = D_0^2$. We now state the next Theorem.

Theorem 2.5.4. *Consider the two $GI/G/K$ queues with resequencing described above. If*

$$A_n^1 =_{st} A_n^2 \qquad n = 0, 1 \ldots (2.82a)$$

and

$$\sigma_n^1 \geq_{s.t.} \sigma_n^2 \qquad n = 0, 1 \ldots (2.82b)$$

then

$$Y_n^1 \geq_{st} Y_n^2 \qquad n = 0, 1 \ldots (2.83)$$

Proof. By the standard construction, define the following sequences of RV's on some fixed probability space $(\Omega, \mathbb{F}, \mathbb{P})$, $\{A_n^1\}_0^\infty$,$\{\sigma_n^1\}_0^\infty$ $\{A_n^2\}_0^\infty$ and $\{\sigma_n^2\}_0^\infty$, such that

$$A_n^1 = A_n^2 \quad \text{and} \quad \sigma_n^1 \geq \sigma_n^2 \qquad n = 0, 1 \ldots \tag{2.84}$$

where (2.84) is always satisfied due to (2.82a-b). Using (2.84), it can be shown as in [29] that

$$D_n^1 \geq D_n^2 \qquad n = 0, 1 \ldots \tag{2.85}$$

from which it follows that

$$Y_n^1 \geq Y_n^2 \qquad n = 0, 1 \ldots \tag{2.86}$$

and (2.83) is now a direct consequence of (2.86).

∎

Theorem 2.5.4 can be used to generate upper bounds in the following way. Assume that the disordering system is a $M/G/K$ queue, and further assume that the service time distribution is NBU with mean μ. From Property (5) of strong stochastic orderings in Appendix B and (3.80c), it follows that the system of this queue is upper bounded by the system time a $M/M/K$ queue with resequencing whose service time also has mean μ. Since the latter system has been solved in [80], the upper bound can be computed.

2.5.4 Variation of System Delay with Number of Servers

We now present a result that generalizes the conclusion of Theorem 2.5.2. Recall that in Theorem 2.5.2 we proved that the system time of a $GI/G/K$ queue with resequencing stochastically upper bounds the system time of a $GI/G/\infty$ queue with resequencing. We now show that the system time of a $GI/G/(K+1)$ queue with resequencing stochastically lower bounds the system time of a $GI/G/K$ queue with resequencing. Hence interestingly enough, increasing the number of servers even by one, causes a decrease in system time. As usual we assume that both systems have identical input processes. For all $n = 0, 1\ldots$ we pose the notation,

a_n^1: Time of arrival of the n^{th} customer into the $GI/G/K$ system.

a_n^2: Time of arrival of the n^{th} customer into the $GI/G/(K+1)$ system.

σ_n^1: Service time of the n^{th} customer *to enter service* in the $GI/G/K$ queue.

σ_n^2: Service time of the n^{th} customer to enter service in the $GI/G/(K+1)$ queue.

u_n^1: Time instant of the n^{th} departure from the $GI/G/K$ queue buffer.

u_n^2: Time instant of the n^{th} departure from the $GI/G/(K+1)$ queue buffer.

v_n^1: Time instant of the n^{th} departure from the servers of the $GI/G/K$ queue.

v_n^2: Time instant of the n^{th} departure from the servers of the $GI/G/(K+1)$ queue.

Note that in case of $GI/G/K$ queues, the n^{th} customer to enter service is also the n^{th} customer to enter the queue, since we assume that the queue operates under the FCFS discipline. Also note that the n^{th} departure from either one of the queues is not necessarily the same as the n^{th} arrival into that queue. Hence a_n^i and v_n^i may describe different customers.

Let $N^1 = K$ and $N^2 = K+1$ in what follows. We now state the main result.

Theorem 2.5.5. *Consider the $GI/G/K$ queue and the $GI/G/(K+1)$ queue with resequencing. If*

$$A_n^1 =_{s.t.} A_n^2 \qquad n = 0, 1\ldots \quad (2.87a)$$

and

$$\sigma_n^1 =_{s.t.} \sigma_n^2 \qquad n = 0, 1\ldots \quad (2.87b)$$

then

$$Y_n^1 \geq_{s.t.} Y_n^2 \qquad n = 0, 1 \ldots \text{(2.88)}$$

Proof. By the standard construction we can define the following sequences of random variables on a fixed probability space $(\Omega, \mathbb{F}, \mathbb{P})$, $\{A_n^1\}_0^\infty$, $\{\sigma_n^1\}_0^\infty$, $\{A_n^2\}_0^\infty$ and $\{\sigma_n^2\}_0^\infty$, such that

$$\begin{aligned} A_n^1 &= A_n^2 \qquad \text{and} \\ \sigma_n^1 &= \sigma_n^2 \end{aligned} \qquad n = 0, 1 \ldots \text{(2.89)}$$

Where (2.89) is always satisfied due to (2.87a-b). From now on, quantities that are common to both systems are written without a super-script. We first prove that departure epochs from the buffer and the servers occur sooner in the $GI/G/(K+1)$ system than in the $GI/G/K$ system. For both systems,

$$\begin{aligned} v_0^i &= \min_{j \geq 0} \{u_j^i + \sigma_j\} \\ &= \min_{0 \leq j < N^i} \{a_j + \sigma_j\} \end{aligned} \qquad \text{(2.90)}$$

and, in general

$$v_j^i = j^{th} \text{ order statistic from } \{u_k^i + \sigma_k : 0 \leq k < j + N^i\} \qquad \text{(2.91)}$$

Since the service initiation of the n^{th} customer coincides with the departure epoch of the $(n - N^i)^{th}$ customer from the system (provided the n^{th} customer arrives before the $(n - N^i)^{th}$ customer has departed the system), the following equation holds

$$u_n^i = \max\{a_n, v_{n-N^i}^i\} \qquad n = 0, 1 \ldots \text{(2.92)}$$

where $v_j^i = 0$ if $j < 0$. We now show that

$$u_n^1 \geq u_n^2 \qquad n = 0, 1 \ldots \text{(2.93)}$$

From (2.92) it follows that $u_n^1 = u_n^2$, for $0 \leq n < leqK$, since the first K customers in either system, do not suffer any queueing delays. However note that

$$u_K^1 = \max\{a_K, v_0^1\} \geq a_K = u_K^2$$

The proof proceeds by induction with an induction step which assumes that for some $n \geq K$,

$$u_j^1 \geq u_j^2 \quad 0 \leq j < n \tag{2.94}$$

From (2.91), it follows that

$$v_{j+1}^1 \geq v_j^2 \quad 0 \leq j \leq n - K - 1 \tag{2.95}$$

and (2.92) now yields

$$\begin{aligned} u_n^1 &= \max\{a_n, v_{n-K}^1\} \\ &\geq \max\{a_n, v_{n-(K+1)}^1\} \\ &= u_n^2 \end{aligned} \tag{2.96}$$

which completes the induction step and the proof of (3.89).

We now obtain an ordering for the total time spent in the queue in the following way. It is clear that if D_n^i is the total time the n^{th} customer spends in the i^{th} queue, then

$$D_n^i = u_n^i - a_n + \sigma_n \qquad n = 0, 1 \ldots \tag{2.97}$$

for every $i = 1, 2$.

From (2.93) and (2.97), it is now clear that

$$D_n^1 \geq D_n^2 \qquad n = 0, 1 \ldots \tag{2.98}$$

From (2.98), using well known techniques, we can prove that

$$Y_n^1 \geq Y_n^2 \qquad n = 0, 1 \ldots \tag{2.99}$$

and (2.88) follows directly from (2.99).

∎

Theorems 2.5.3 and 2.5.5 reveal an interesting structural feature of multiple server resequencing systems. Theorem 2.5.5 states that the system delay decreases if we add an additional server to the multiserver system. However note that the resequencing delay clearly does not decrease because more customers may go out of sequence as result of the presence of the additional server. Hence the crux of Theorem 2.5.5 is that the decrease in queueing delay due the presence of the additional server, outweighs the increase in synchronization delay due to the resequencing constraint. Hence it is all-right to increase the amount of parallelism in the system as much as possible without worrying about resequencing delays. Theorem 2.5.3 states that this property also holds in the limit as the number of servers goes to infinity. An interesting open problem is to charectarize the behaviour of the resequencing delay as the number of servers is increased. Clearly, since it increases as more servers are added and yet does not go to infinity in an infinite server system, its distribution must converge to stable distribution at infinity. This situation is in direct contrast to the behaviour of a fork-join queue, whose system time increases logarithmically with the number of servers in the queue [7].

CHAPTER III

RESEQUENCING IN MULTISTAGE DISORDERING SYSTEMS

3.1 Introduction

In this chapter we identify several properties of multi-stage disordering systems with resequencing. The short survey of the literature given in Chapter 1, revealed a paucity of results concerning multistage resequencing systems, which is not surprising considering their extremely complex nature. However, as shown in this chapter, sometimes interesting properties of these systems can be deduced by using stochastic comparison techniques.

Chapter 3 is organized as follows. In Section 3.2 we prove that hop-by-hop resequencing stochastically upper bounds end-to-end resequencing, in a system consisting of a general disordering queue followed by a $G/G/\infty$ queue in tandem. We also extend the result to an arbitrary number of $G/G/\infty$ queues in tandem. In Section 3.3 we present a number of structural results concerning hop-by-hop resequencing systems, in particular we show that most of the structural results about $G/G/K$ queues obtained in Section 2.5, also extend to system of N $G/G/K$ queues in tandem with hop-by-hop resequencing. In Section 3.4 we prove the stability of the distribution of a two hop end-to-end resequencing delay, thus extending the results in [Baccelli, Gelenbe, Plateau] where stability was proved for a single hop resequencing system. We also a derive an integral equation for this distribution.

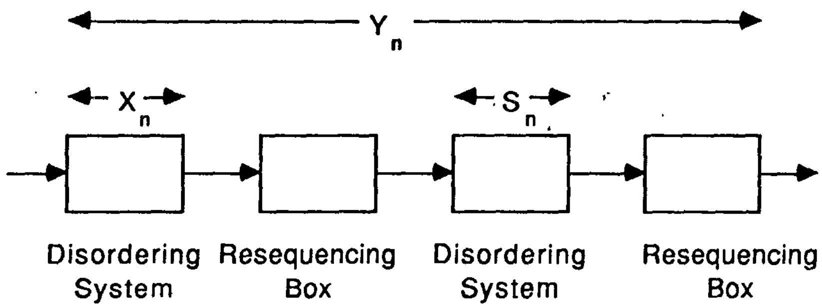

Fig 3.2.1 (a). Hop-by-Hop Resequencing System

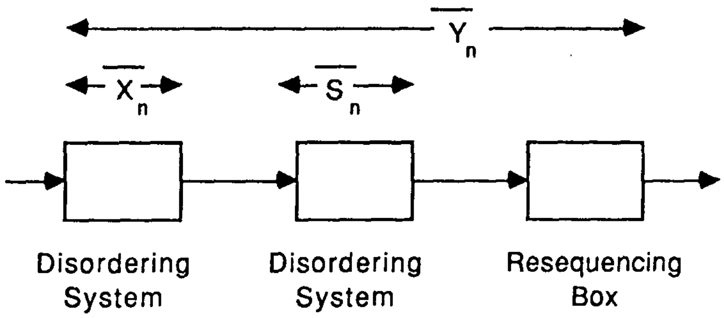

Fig 3.2.1 (b). End-to-End Resequencing System

3.2 The Optimality of End-to-End Resequencing

Given a multi-stage disordering system, a probem of considerable interest is the effect of various resequencing strategies on system delay. Yum and Ngai [80], presented simulation results on the comparison of resequencing delays for the two kinds of resequencing strategies in a two hop disordering system (Fig 3.2.1). The disordering in both stages was carried out by $M/M/K$ queues. In the first case, resequencing was done after a customer had traversed both queues, while in the second case, resequencing was implemented after each queue. We shall hereafter refer to the first strategy as *end-to-end* resequencing, and to the second strategy as *hop-by-hop* resequencing. The simulation results showed that the average hop-by-hop resequencing delay was greater than the average end-to-end resequencing delay for two stage disordering systems.

In the present section we shall compare different kinds of resequencing strategies in tandem systems, when the disordering is due to infinite server queues. Our results are stronger than the simulation results in [80] in two respects. The ordering we get is strict sample path ordering for each customer, and secondly it holds for any number of disordering stages. However we have been able to prove the result only for infinite server queues.

The discussion starts with the two hop resequencing systems depicted in Figs 3.2.1 (a) and 3.2.1 (b) above. For all $n = 0, 1 \ldots,$ pose

$\overline{Y}_n$: Delay of the n^{th} customer in the end-to-end resequencing system.

Y_n: Delay of the n^{th} customer in the hop-by-hop resequencing system.

X_n: Delay of the n^{th} customer in the first disordering system of the hop-by-hop resequencing tandem system.

$\overline{X}_n$: Delay of the n^{th} customer in the first disordering system of the end-to-end resequencing tandem system.

A_{n+1}: Inter-arrival time the $(n+1)^{rst}$ and the n^{th} customers in the hop-by-hop tandem resequencing system.

$\overline{A}_{n+1}$: Inter-arrival time between the $(n+1)^{rst}$ and the n^{th} customer in the end-to-end resequencing tandem system.

S_n: Delay of the n^{th} customer in the second disordering system of the hop-by-hop resequencing tandem system.

$\overline{S}_n$: Delay of the n^{th} customer in the second disordering system of the end-to-end resequencing system.

D_n: Delay of the n^{th} customer in the hop-by-hop resequencing scheme, due to the two disordering systems and the first resequencing box.

Z_n: Delay of the n^{th} customer in the hop-by-hop resequencing system, due to the first disordering stage and the first resequencing box.

Since our aim is to understand how the system times vary with the resquencing strategy, we assume that the two disordering systems and the inter-arrival time statistics are identical in both cases. More precisely, if all these sequences are defined over some common sample space $(\Omega, \mathbb{F}, P)$, then the following equations

$$A_n = \overline{A}_n \qquad n = 0, 1 \ldots (3.1a)$$

$$X_n = \overline{X}_n \qquad n = 0, 1 \ldots (3.1b)$$

$$S_{n+1} = \overline{S}_{n+1} \qquad n = 0, 1 \ldots (3.1c)$$

are assumed to hold. Note that in writing (3.1), we have made a subtle assumption which restricts the class of disordering systems considered here. Condition $S_n = \overline{S}_n$, does not hold true in general as we now show. Consider the situation where the second disordering system is a $G/G/K$ queue, in which case the n^{th} customer to enter the system may undergo diferent queueing delays at this queue, depending on whether the resequencing is done hop-by-hop or end-to-end. This is because of the first resequencing box which drastically changes the nature of the arrival process into the $G/G/K$ queue. Hence (3.1c) is applicable only to those systems in which the *second* disordering delay is not affected by the arrival process into it. One class of disordering systems to which this is applicable, is the class of systems having an infinite number of servers provided the delays in this system are generated independently of the arrival process $\{A_n\}_0^\infty$ as well as the delays $\{X_n\}_0^\infty$ in the first disordering system. In all the results presented in this section, we shall restrict ourselves to this case.

Theorem 3.2.1. *Consider a two stage disordering system with resequencing, in which the second stage has an infinite number of servers, then the system delay for the end-to-end resequencing system is stochastically upper bounded by the system delay of the hop-by-hop resequencing system, i.e.,*

$$\overline{Y}_n \leq_{st} Y_n. \qquad n = 0, 1 \ldots (3.2)$$

Proof. From the statement of the theorem we can assume that (3.1) holds on some probability space $(\Omega, \mathbb{F}, P)$.

First consider the end-to-end resequencing system. Application of the basic Theorem 2.1.1 from Chapter 2 gives

$$\overline{Y}_{n+1} = \max\{X_{n+1} + S_{n+1}, \overline{Y}_n - A_{n+1}\} \qquad n = 0, 1 \ldots (3.3)$$

for the end-to-end resequencing system, and

$$Y_{n+1} = \max\{D_{n+1}, Y_n - A_{n+1}\} \qquad n = 0, 1 \ldots (3.4)$$

for the hop-by-hop resequencing system.A

We now derive a recursive expression for the sequence $\{D_n\}_0^\infty$. Application of Theorem 2.1.1 to the first disordering stage followed by resequencing in the hop-by-hop resequencing system yields

$$Z_{n+1} = \max\{X_{n+1}, Z_n - A_{n+1}\} \qquad n = 0, 1 \ldots (3.5)$$

with $Z_0 = X_0$. Since

$$D_n = Z_n + S_n \qquad n = 0, 1 \ldots (3.6)$$

it follows from (3.5) that

$$D_{n+1} = S_{n+1} + \max\{X_{n+1}, D_n - S_n - A_{n+1}\} \qquad n = 0, 1 \ldots (3.7)$$

with $D_0 = S_0 + X_0$.

Next we use induction to prove that

$$\overline{Y}_n \leq Y_n, \qquad n = 0, 1 \ldots (3.8)$$

in which case (3.2) immediately follows.

For $n = 0$, under the zero initial loading assumption in both systems, it is plain that

$$Y_0 = D_0 = S_0 + X_0 = \overline{Y}_0,$$

whence, (3.8) is satisfied for the 0^{th} customer. The induction step assumes that (3.8) holds for the m^{th} customer so that

$$\overline{Y}_m \leq Y_m \qquad (3.9)$$

or equivalently,

$$\overline{Y}_m - \overline{A}_{m+1} \leq Y_m - A_{m+1} \qquad (3.10)$$

Since the inequality

$$\overline{X}_{m+1} \leq \max\{X_{m+1}, D_m - S_m - A_{m+1}\} \tag{3.11}$$

always holds, it follows that

$$\overline{S}_{m+1} + \overline{X}_{m+1} \leq S_{m+1} + \max\{X_{m+1}, D_m - S_m - A_{m+1}\} \tag{3.12}$$

by (3.7), this is equivalent to

$$\overline{S}_{m+1} + \overline{X}_{m+1} \leq D_{m+1} \tag{3.13}$$

By combining (3.10) and (3.13), it follows that

$$\max\{\overline{Y}_m - \overline{A}_{m+1}, \overline{S}_{m+1} + \overline{X}_{m+1}\} \leq \max\{Y_m - A_{m+1}, D_{m+1}\} \tag{3.14}$$

and we now easily obtain from (3.3)-(3.4) that

$$\overline{Y}_{m+1} \leq Y_{m+1}$$

i.e., (3.8) holds for $n = m + 1$. Since equation (3.8) holds for $n = 0$, it follows by induction that it is true for all $n = 0, 1 \ldots$.

∎

Note that the assumptions of Theorem 3.2.1 can be weakened to

$$\begin{aligned} A_n &\leq \overline{A}_n \\ X_n &\geq \overline{X}_n \qquad n = 0, 1 \ldots \\ S_n &\geq \overline{S}_n \end{aligned} \tag{3.15}$$

without changing the conclusion of the theorem. The reader might expect that in order to extend the theorem to the case when the second disordering system has a finite number of servers, it is suffucient to verify that

$$S_n \geq \overline{S}_n \qquad n = 0, 1 \ldots \tag{3.16}$$

However there is some reason to believe that (3.16) might not hold for this case, as intuitively argued below.

Let us consider a system in which the disordering is carried out by a $GI/G/2$ queue in both stages. Consider the n^{th} customer C_n which is in the process of receiving service from one of the servers in the first $GI/G/2$ queue in the hop-by-hop resequencing system. If his service time is inordinately long, then customers who had arrived after him into the first queue, will complete their service before him and wait in the resequencing box for C_n to complete service. Assume that customers C_{n+1} to C_{n+k} have gone out of sequence with respect to C_n, and are waiting in the resequencing box for C_n. After C_n completes service, he will immediately join the second queue (before C_{n+1} to C_{n+k}), since he does not suffer any resequencing delay in the first resequencing box. On the other hand, for the case of end-to-end resequencing, C_{n+1} to C_{n+k} would immediately join the buffer of the second queue after getting served in the first queue. Consequently, after C_n finishes service in the first queue he would find a bigger queue length in the second queue, than for the case of hop-by-hop resequencing. Thus we would expect the delay of C_n in the end-to-end resequencing system to be greater than his delay in the hop-by-hop resequencing system. The delays of C_{n+1} to C_{n+k} though would be smaller in the end-to-end resequencing system. Hence it is not unlikely that the average delay for end-to-end resequencing is smaller than the average delay for hop-by-hop resequencing, even though sample path ordering is not possible, in fact *false* and comparison results can only be expected in a weaker sense.

Corollary 3.2.1 extends the result of Theorem 3.2.1 to any number, say N, of infinite server queues in tandem. For all $n = 0, 1, \ldots$, pose

S_n^N: Delay of the n^{th} customer due to the N^{th} disordering stage in the hop-to-hop resequencing system.

$\overline{S}_n^N$: Delay of the n^{th} customer due to the N^{th} disordering stage in the end-by-end resequencing system.

Y_n^N: delay of the n^{th} customer in the hop-by-hop resequencing system consisting of N disordering stages.

$\overline{Y}_n^N$: delay of the n^{th} customer in the end-to-end resequencing system consisting of N disordering stages.

Corollary 3.2.1. *Given N (≥ 2) $GI/G/\infty$ queues in tandem, the hop-by-hop resequencing delay for a customer is stochastically no smaller than the end-to-end resequencing delay for that customer, i.e.,*

$$\overline{Y}_n^N \leq_{st} Y_n^N \qquad n = 0, 1 \ldots \text{(3.17)}$$

Proof. Since we are only interested in comparing system times due to the difference in resequencing strategies, we will assume that

$$\overline{S}_n^N = S_n^N \qquad n, N = 0, 1 \ldots \text{(3.18}a\text{)}$$

$$\overline{A}_n = A_n \qquad n = 0, 1 \ldots \text{(3.18}b\text{)}$$

The proof proceeds by a double induction on the number of customers as well as the number of stages. By Theorem 3.2.1, it is clear that that (4.16) is true for $N = 2$. We will show that (4.16) holds for an arbitrary value of $N = M$.

Applying Theorem 2.2.1, we obtain the relations

$$Y_{n+1}^M = \max\{Y_{n+1}^{M-1} + S_{n+1}^M, Y_n^M - A_{n+1}\} \qquad n = 0, 1 \ldots \text{(3.19)}$$

and

$$\overline{Y}_{n+1}^M = \max\{\sum_{i=1}^{M} S_{n+1}^i, \overline{Y}_n^M - A_{n+1}\} \qquad n = 0, 1 \ldots \text{(3.20)}$$

We have to show that

$$\overline{Y}_n^M \leq Y_n^M \qquad n = 0, 1 \ldots \text{(3.21)}$$

Assume that (3.21) is true for $n = m$

$$\overline{Y}_m^M \leq Y_m^M \qquad \text{(3.22)}$$

which implies that

$$\overline{Y}_m^M - A_{m+1} \leq Y_m^M - A_{m+1} \tag{3.23}$$

Next by an induction on the nunber of disordering stages, we demonstrate that

$$\sum_{i=1}^{M-1} \overline{S}^i_{m+1} \leq Y^{M-1}_{m+1} \tag{3.24}$$

From (3.11), it is clear that (3.24) holds for $M = 2$. As the induction step assume that (3.24) holds for $M = L$, i.e.,

$$\sum_{i=1}^{L-1} \overline{S}^i_{m+1} \leq Y^{L-1}_{m+1} \tag{3.25}$$

It is clear that

$$Y^L_{m+1} = \max\{Y^{L-1}_{m+1} + S^L_{m+1}, Y^L_m - A_{m+1}\} \tag{3.26}$$

Hence it is immediate to see that

$$Y^L_{m+1} \geq Y^{L-1}_{m+1} + S^L_{m+1} \geq \sum_{i=1}^{L-1} \overline{S}^i_{m+1} + \overline{S}^L_{m+1}$$

$$= \sum_{i=1}^{L} \overline{S}^i_{m+1} \tag{3.27}$$

Hence (3.25) holds for $M = L + 1$ as well, completing the induction step. Hence from (3.23-24), it follows that

$$\overline{Y}^M_{m+1} \leq Y^M_{m+1} \tag{3.28}$$

thus completing the indution step and the proof.

∎

In addition to being an useful qualitative result, Corollary 3.2.1 can be used to generate numerical bounds as we now explain. Consider a N stage disordering system, where each stage corresponds to a $GI/G/\infty$ queue. The hop-by-hop resequencing delay in this system is analytically intractable. However by Corollary 3.2.1, it is bounded from below by the end-to-end resequencing delay in the N-stage system, in the sense of strong stochastic ordering. It is well known that the strong stochastic ordering has the weak convergence property (see Appendix B). Hence if the random variables Y_n^N and $\overline{Y}_n^N$, converge in distribution to Y_∞^N and $\overline{Y}_\infty^N$ respectively (this is proved rigorously in the next section for two stage disordering systems), then from (4.2)

$$E[f(\overline{Y}_\infty^N)] \leq E[f(Y_\infty^N)] \tag{3.29}$$

for all non-decreasing functions f. All moments of Y_∞^N are thus bounded from below by the corresponding moment of $\overline{Y}_\infty^N$. and in particular

$$E[Y_\infty^N] \geq E[\overline{Y}_\infty^N]. \tag{3.30}$$

Denoting the delay due to the N disordering systems in the end-to-end disordering system as $\overline{D}_\infty^N$, we can write

$$\overline{D}_\infty^N = \sum_{i=1}^{N} S^i \tag{3.31}$$

where S^i has the same distribution as that of the i^{th} disordering system. Denoting the distribution of D_∞^N as F, and the distribution of S^i as $F_i, 1 \leq i \leq N$, (3.31) implies

$$F = F_1 * F_2 * \ldots * F_N \tag{3.32}$$

Assume now that each distribution $F_i, 1 \leq i \leq N$ is an exponential distribution with parameter μ, and the inter-arrival times in the first disordering system are exponentially distributed with parameter λ. It follows that $\overline{D}_\infty^N$ is distributed according to an Erlang distribution of order N, with density given by

$$f_{\overline{D}_\infty^N}(x) = \frac{\mu(\mu x)^{N-1}}{(N-1)!}\exp(-\mu x), \quad x \geq 0 \tag{3.33}$$

Hence the N-stage end-to-end disordering system is equivalent to a single stage disordering system whose disordering delay is Erlang distributed, in other words a $M/Er(N)/\infty$ queue with resequencing. According to [23], if we denote by F the distribution of the disordering delay in a $M/G/\infty$ queue with resequencing, then $E[\overline{Y}]$ can be written as

$$E[\overline{Y}_{\infty}^{N}] = E[\overline{D}_{\infty}^{N}] + \int_{0}^{\infty} (1 - \exp^{-\lambda \int_{0}^{t} (1-F(u))du}) F(t) dt \tag{3.34}$$

In our case $E[\overline{D}_{\infty}^{N}] = N/\mu$ and F has density given by (3.33). The second expression on the right hand side of (3.34) can be evaluated, perhaps numerically, after substituting the Erlang distribution in place of F. This gives us a computable lower bound to the average system delay of the N-stage hop-by-hop resequencing system.

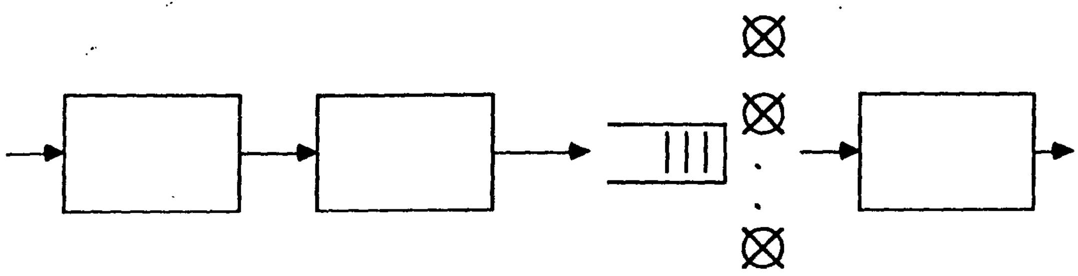

Fig 3.3.1. The Resequencing System

3.3 Finite Server Queues in Tandem with Resequencing

Consider a system consisting of a $GI/G/K_1$ queue in tandem with a $GI/G/K_2$ queue (without resequencing and with $K_1, K_2 \geq 2$). If there is an increase in the number of servers or the service rate at the first $GI/G/K_1$ queue, then classical results due to Jacobs and Schac [29], tell us that the system time of a customer decreases sample-pathwise in that queue. However this decrease does not carryover to the end-to-end delay of a customer due to both the queues, in other words a decrease in system time at the first queue does not imply a decrease in end-to-end delay [58]. However, in this section we show that if the customers are resequenced after each queue, then a decrease in system time at the first queue does imply a decrease in the end-to-end delay. This property is further extended to an arbitrary number of $GI/G/K$ queues in tandem, with resequencing after each stage. However it does not seem to apply to end-to-end resequencing systems.

The model is now introduced with the appropriate notations. The first disordering system is allowed to be arbitrary, (Fig 3.3.1) while the second disordering system is assumed to be a $GI/G/K$ queue. Resequencing is carried out after each disordering stage. In the next theorem we show that a decrease in the system time at the first disordering system, implies a samplepathwise decrease in theend-to-end delay. In what follows we use the super-script $i = 1, 2$to refer to the two systems. For all $n = 0, 1 \ldots$, and $i = 1, 2$ pose,

Y_n^i: End-to end delay of the n^{th} customer in the i^{th} system.

A_{n+1}^i: Inter-arrival time between the $(n+1)^{rst}$ and the n^{th} customer into the i^{th} system.

X_n^i: Delay of the n^{th} customer in the first disordering system in the i^{th} system.

Z_n^i: Delay of the n^{th} customer in the i^{th} system due to the first disordering system and the following resequencing box.

a_n^i: Arrival instant of the n^{th} customer in the i^{th} system.

τ_n^i: Arrival instant into the $GI/G/K$ queue, of the n^{th} customer to enter the i^{th} system.

J_n^i: Departure instant from the $GI/G/K$ queue buffer of the n^{th} customer to enter the i^{th} system.

d_n^i: departure instant from the system of the n^{th} customer to enter the i^{th} system.

$\overline{\sigma}_n^i$: service time in the $GI/G/K$ queue, of the n^{th} customer to enter the i^{th} system.

We now state the main result in this section.

Theorem 3.3.1. *Consider the two double stage disordering systems, in which the first system is allowed to be arbitrary, while the second system corresponds to a $GI/G/K$ queue. If on some sample space $(\Omega, \mathbb{F}, P)$, the following relations hold*

$$\begin{aligned} A_{n+1}^1 &= A_{n+1}^2 \\ X_n^1 &\geq X_n^2 \qquad n = 0, 1 \ldots \quad (3.35) \\ \sigma_n^1 &= \sigma_n^2 \end{aligned}$$

then the inequalities

$$Y_n^1 \geq_{st} Y_n^2 \qquad n = 0, 1 \ldots \quad (3.36)$$

hold true.

Proof. It is plain from (3.35) that

$$\tau_n^1 \geq \tau_n^2 \qquad n = 0, 1 \ldots (3.37)$$

since by defenition

$$\tau_n^i = a_n^i + X_n^i \qquad n = 0, 1 \ldots (3.38)$$

for all $i = 1, 2$.

We now focus our attention on the $GI/G/K$ queue. The first thing to note is that

$$J_n^1 \geq J_n^2 \qquad n = 0, 1 \ldots (3.39)$$

A little thought will convince the reader that (3.39) follows directly from (3.35) and (3.37) since the order in which customers are sent into service in the $GI/G/K$ queue is the same as the order in which they entered it. However this statement does not hold for K $G/G/1$ queues in parallel.

Our next step is to show that

$$d_n^1 \geq d_n^2 \qquad n = 0, 1 \ldots (3.40)$$

To that end, using the basic recursion of Theorem 2.2.1, it is not very difficult to see that;

$$d_{n+1}^i = \max\{d_n^i, J_n^i + \sigma_n^i\} \qquad n = 0, 1 \ldots (3.41)$$

for $i = 1, 2$. We now prove (3.40) by induction. It is plain from (3.35) that

$$d_0^1 = X_0^1 + \sigma_0^1 \geq X_0^2 + \sigma_0^2 = d_0^2 \qquad (3.42)$$

and (3.40) is thus satisfied for $n = 0$. Assume that (3.40) is true for some $n = m \geq 0$, i.e.,

$$d_m^1 \geq d_m^2 \qquad (3.43)$$

it immediately follows

$$d_{m+1}^1 \geq d_{m+1}^2 \qquad (3.44)$$

and this completes the induction.

Since

$$Y_n^i = d_n^1 - a_n^i \qquad n = 0, 1 \ldots (3.45)$$

for $i = 1, 2$, it directly follows that (3.36) holds. ∎

The next corollary follows directly from the above result and Theorems 2.3-2.5 from Chapter 2.

Corollary 3.3.1. *Consider a two stage disordering system, with hop-by-hop re-sequencing, in which the first stage is a $GI/G/K_1$ queue and the second stage is $GI/G/K_2$ queue. If any of the following changes (i)-(iii) are made to the system, where*

(i) The number of servers in either or both disordering systems is increased to $K_j + k_j, k_j \geq 0$, while keeping the service distribution of the additional servers the same as those of the original servers,

(ii) Either or both the disordering systems are replaced by an infinite server system whose service distribution is the same as for the original systems,

(iii) The service processes in either or both the disordering systems is changed to $\{\sigma_{nj}^2\}_0^\infty, j = 1, 2$, such that

$$\sigma_{nj}^2 \geq \sigma_{nj}^1 \quad j = 1, 2, \qquad n = 0, 1 \ldots (3.46)$$

then

$$Y_n^1 \geq_{st} Y_n^2 \qquad n = 0, 1 \ldots (3.47)$$

Proof. Consider the case when the $G/G/K_1$ queue is altered. For $i = 1, 2$, let X_n^i denote the system delay in this queue of the n^{th} customer to enter it, before and after alteration. It is clear from Theorems 2.5.3-2.5.5, that in each of the cases (i)-(iii) above, the following equation

$$X_n^1 \geq X_n^2 \qquad n = 0, 1 \ldots (3.48)$$

is satisfied. Hence the conditions of Theorem 3.3.1 are satisfied in this case, so that (3.47) follows directly from (3.36).

Now consider the case when the $G/G/K_2$ queue is altered. In this case

$$A_n^1 = A_n^2 \quad \text{and} \quad X_n^1 = X_n^2 \qquad\qquad n = 0, 1 \ldots$$

For cases (i) and (ii), (3.47) follows from the fact that $J_n^1 \geq J_n^2$ for all $n = 0, 1 \ldots$ while for case (iii), (3.47) follows from the fact that $\sigma_n^1 \geq \sigma_n^2$ for all $n = 0, 1 \ldots$. The details are left to the interested reader.

∎

The next result extends Corollary 3.3.1, to any number N, of multi-server queues in tandem.

Corollary 3.3.2. *Consider a N stage disordering system, with hop-by-hop resequencing, in which the i^{th} stage corresponds to a $G/G/K_i$ queue $1 \leq i \leq N$. If any of the following changes are made, where*

(i) The number of servers in the i^{th} queue is increased to $K_i + k_i$ with $k_i \geq 0, \leq i \leq N$, while keeping the service distribution of the additional servers the same as those of the original servers,

(ii) The i^{th} queue, $1 \leq i \leq N$, is replaced by an infinite server queue whose service distribution is the same as for the original queue,

(iii) The service process in the i^{th} queue, $1 \leq i \leq N$, is changed to $\{\sigma_{ni}^2\}_0^\infty$, such that

$$\sigma_{ni}^2 \geq \sigma_{ni}^1, \qquad\qquad n = 0, 1 \ldots \quad (3.49)$$

then

$$Y_n^1 \geq_{st} Y_n^2 \qquad\qquad n = 0, 1 \ldots \quad (3.50)$$

Proof. A short sketch of the proof is provided, the details of which are left to the reader.

Suppose that the i^{th} queue is altered. Then the delay of a customer in the first $(i-1)$ queues is unchanged, however the departure epoch of a customer from the i^{th} queue will be earlier in the altered system. This in turn implies that the departure epoch of that customer will be earlier in each of the downstream queues, and ultimately the system, for the altered case. This can be proved in exactly the same way as Theorem 3.3.1, and implies (3.50).

∎

3.4 Stability of the Two stage Disordering System

Consider a two stage disordering system with end-to-end resequencing. Assume that the disordering in both stages is carried out by $GI/G/\infty$ queues. Our goal in this section is to prove that the end-to-end system delay Y_n converges in distribution to a proper random variable Y_∞. This constitutes an extension of the results in [7], where stability of a single stage resequencing system was proved.

We will use the same notation for the end-to-end resequencing queue as in Theorem 3.2.1. The main result is contained in Theorem 3.4.1, in which we prove a somewhat stronger result, i.e., the RV's (D_n, Y_n) jointly converge in distribution to (D_∞, Y_∞). This will be essential in developing the integral equation for which the joint equlibrium distribution of (D_∞, Y_∞), which is done subsequently.

Recall that the sequences $\{D_n\}_0^\infty$ and $\{Y_n\}_0^\infty$ where recursively defined by the equations

$$D_{n+1} = \max\{X_{n+1} + S_{n+1}, D_n + S_{n+1} - S_n - A_{n+1}\} \quad \text{with } D_0 = S_0 + X_0 \qquad n = 0, 1 \ldots \tag{3.51}$$

and

$$Y_{n+1} = \max\{D_{n+1}, Y_n - A_{n+1}\} \quad \text{with } Y_0 = D_0 \qquad n = 0, 1 \ldots \tag{3.52}$$

We will make the following assumption (C1), where

(C1) The RV's $\{S_n\}_0^\infty$, $\{X_n\}_0^\infty$ and $\{A_n\}_1^\infty$ are mutually independent sequences of i.i.d RV's with finite means, i.e.,

$$E[S_n] < \infty \qquad E[X_n] < \infty \qquad E[A_n] < \infty \qquad n = 0, 1 \ldots$$

We now state the main result in this seection.

Theorem 3.4.1. *Under the assumption (C1), the sequence $\{D_n, Y_n\}_0^\infty$ converges in distribution to finite RV's (D_∞, Y_∞).*

Proof. By iterating on (3.51)-(3.52) it follows that for all $n = 0, 1 \ldots$,

$$D_n = \max\{S_n + X_n, S_n + X_{n-1} - A_n, \ldots, S_n + X_0 - A_n - \ldots - A_1\} \tag{3.53}$$

and

$$Y_n = \max\{D_n, D_{n-1} - A_n, \ldots, D_0 - A_n - \ldots - A_1\} \qquad n = 0, 1, \ldots \quad (3.54)$$

Substituting for $D_m, m = 0, \ldots, n$ from (3.53) in (3.54) we obtain

$$\begin{aligned}(D_n, Y_n) = (&\max\{S_n + X_n, S_n + X_{n-1} - A_n, \ldots, S_n + X_0 - A_n - \ldots - A_1\},\\ &\max\{S_n + X_n, S_n + X_{n-1} - A_n, \ldots, S_n + X_0 - A_n - \ldots - A_1,\\ &\quad S_{n-1} + X_{n-1} - A_n, \ldots, S_{n-1} + X_0 - A_n - \ldots - A_1,\\ &\quad \vdots\\ &\quad \ldots, S_0 + X_0 - A_n - \ldots - A_1\})\end{aligned}$$

$$n = 0, 1 \ldots \quad (3.55)$$

Following Loynes [48], we imbed the sequences $\{S_n\}_0^\infty$, $\{X_n\}_0^\infty$ and $\{A_n\}_1^\infty$ into the larger stationary ergodic sequences $\{S_n\}_{-\infty}^\infty$, $\{X_n\}_{-\infty}^\infty$ and $\{A_n\}_{-\infty}^\infty$ respectively. The RV's $\{D'_n, Y'_n\}_0^\infty$ are now defined componentwise by

$$\begin{aligned}(D'_n, Y'_n) = (&\max\{S_0 + X_0, S_0 + X_{-1} - A_0, \ldots, S_0 + X_{-n} - A_0 - A_{-(n-1)}\},\\ &\max\{S_0 + X_0, S_0 + X_{-1} - A_0, \ldots, S_0 + X_{-n} - A_0 - \ldots - A_{-(n-1)},\\ &\quad S_{-1} + X_{-1} - A_0, \ldots, S_{-1} + X_{-n} - A_0 - \ldots - A_{-(n-1)},\\ &\quad \vdots\\ &\quad \ldots S_{-n} + X_{-n} - A_0 - \ldots - A_{-(n-1)}\})\end{aligned}$$

$$n = 0, 1, \ldots \quad (3.56)$$

Due to (C1), the RV's (D_n, Y_n) and (D'_n, Y'_n) have the same distribution for all $n = 0, 1 \ldots$.

Note that the sequence $\{D'_n, Y'_n\}_0^\infty$ is non-decreasing in n, and therefore $\lim(D'_n, Y'_n)$ exists. Denote this limiting RV by (D'_∞, Y'_∞). The RV (D'_∞, Y'_∞) is a.s. finite since by (C1), the strong law of large numbers yields,

$$\lim_n \frac{1}{n}(A_0 + \ldots + A_n) = E[A] > 0 \qquad \text{a.s.} \qquad (3.57)$$

$$\lim_{n} \frac{X_n}{n} = 0 \qquad \text{a.s.} \tag{3.58}$$

and

$$\lim_{n} \frac{S_{-n}}{n} = 0 \qquad \text{a.s.} \tag{3.59}$$

Thus, there exists an a.s. finite integer valued RV M_0 such that for all $n > M_0$,

$$S_{-m} + X_{-n} - A_0 - \ldots - A_{-(n-1)} < 0 \qquad 0 \leq m \leq n \tag{3.60}$$

Hence D'_∞ and Y'_∞ are the maximum of an a.s. finite number of proper RV's hence each is itself a proper random variable. Consequently the RV's $\{D_n, Y_n\}_0^\infty$ necassarily converge weakly to an a.s. finite RV (D_∞, Y_∞) which is identical in distribution to the non-defective RV's (D'_∞, Y'_∞).

∎

An integral equation for the joint distribution of (D_∞, Y_∞) is developed next.

From (3.52), it is plain that the sequence $\{Y_n\}_0^\infty$ does not form a Markov Chain because $\{D_n\}_0^\infty$ is not an i.i.d sequence. However the RV's $\{(Y_n, D_n)\}_0^\infty$ do form a two dimensional Markov chain, as can easily be checked from (3.51)-(3.52). We will use this property in deriving the integral equation.

Define for each $n = 0, 1 \ldots$,

$$F_{Y_n, D_n}(y, d) = P[Y_n \leq y, D_n \leq d] \qquad y, d \geq 0$$

Also for convenience pose for each $n = 0, 1 \ldots$,

$$F_A(a) = P[A_n \leq a] \qquad F_X(x) = P[X_n \leq x]$$

$$F_S(s) = P[S_n \leq s]$$

with $a \geq 0, x \geq 0$ and $s \geq 0$. From (3.51)-(3.52) we can write (for $y, d \geq 0$),

$$F_{Y_{n+1}, D_{n+1}}(y, d)$$

$$= \int_0^\infty \int_0^\infty P(Y_{n+1} \leq y, D_{n+1} \leq d \mid Y_n = x, D_n = z)\, dF_{Y_n, D_n}(x, z)$$

$$= \int_0^\infty \int_0^\infty P(D_{n+1} \le y, x - A_{n+1} \le y, D_{n+1} \le d \mid Y_n = x, D_n = z)\, dF_{Y_n,D_n}(x,z)$$

$$= \int_0^\infty \int_0^\infty P(D_{n+1} \le \min(y,d), A_{n+1} \ge x - y \mid Y_n = x, D_n = z)\, dF_{Y_n,D_n}(x,z) \tag{3.61}$$

Let $p = \min(y,d)$, $p' = p - a$, $q = x - y$ and $r = \max(q, z - b + a - p)$. Conditioning the integrand on S_{n+1} and S_n, we get

$$P(D_{n+1} \le p, A_{n+1} \ge q \mid Y_n = x, D_n = z)$$

$$= \int\int P(D_{n+1} \le p, A_{n+1} \ge q \mid Y_n = x, D_n = z, S_{n+1} = a, S_n = b)$$

$$dF_S(a) dF_S(b)$$

$$= \int\int P(X_{n+1} \le p', A_{n+1} \ge r \mid Y_n = x, D_n = z, S_{n+1} = a, S_n = b)$$

$$dF_S(a) dF_S(b) \tag{3.62}$$

where both the integrals are from 0 to infinity. Since X_{n+1} and A_{n+1} are independent of Y_n, D_n, S_{n+1} and S_n, we can re-write the last equation as

$$P(D_{n+1} \le p, A_{n+1} \ge p \mid Y_n = x, D_n = z) =$$

$$\int_0^\infty \int_0^\infty P(X_{n+1} \le p') P(A_{n+1} \ge r)\, dF_S(a) dF_S(b) \tag{3.63}$$

The integral in (3.63) can be evaluated after substituting for the distribution of S_{n+1}, S_n, X_{n+1} and A_{n+1}. Denote its value by $\phi(y,d,x,z)$, so that

$$\phi(y,d,x,z) = \int_0^\infty \int_0^\infty F_X(p')\{1 - F_A(r)\}\, dF_S(a)\, dF_S(b)$$

Hence we can write (3.61) as

$$F_{Y_{n+1},D_{n+1}}(y,d) = \int_0^\infty \int_0^\infty \phi(y,d,x,z)\, dF_{Y_n D_n}(x,z) \tag{3.64}$$

From Theorem 3.4.1 the distribution F_{Y_n,D_n} has a weak limit $F_{Y,D}$, and therefore we have the following integral equation, satisfied by this limiting distribution.

$$
\begin{aligned}
F_{Y,D}(y,d) &= \int_0^\infty \int_0^\infty \phi(y,d,x,z)\, dF_{Y,D}(x,z) \qquad y,d \geq 0 \\
&= 0 \qquad \text{otherwise}
\end{aligned}
\tag{3.65}
$$

CHAPTER IV

THE TWO SERVER RESEQUENCING SYSTEM

4.1 Introduction

In this chapter we present a detailed analysis of the $M/M/2/B$ queue with resequencing. There are a number of reasons why an exact analysis of simple models such as this one is of interest, not the least among which is that perfomance measures of complicated systems can be bounded by those of simpler systems (as amply illustrated elsewhere in the thesis).

The mathematical technique is from Markov chain theory rather than from sample path analysis, in contrast with the material of the last two chapters. The main ingredient required for a Markov chain analysis is a state space representation of the system. As pointed out earlier, the state space representation of most resequencing systems, indeed of most systems with synchronization constraints, is so complex that a Markov chain analysis is all but impossible. However, because of the structure of the $M/M/2/B$ system with resequencing, a state space representation which is simple enough, in fact exists, and can be used to advantage.

The rest of the chapter is organized as follows. In Section 4.2 we give a Markovian state space description of the model. In Section 4.3 we present the corresponding equations for the steady-state probabilities. In Section 4.4 we give an exact solution to these equations for the special case of $B = 0$. In Section 4.5, by using matrix-geometric techniques, we solve them for an arbitrary yet finite value of B. In Section 4.6 we derive formulae for the bulk departure distribution for the cases $B = 0$ and $B = \infty$.

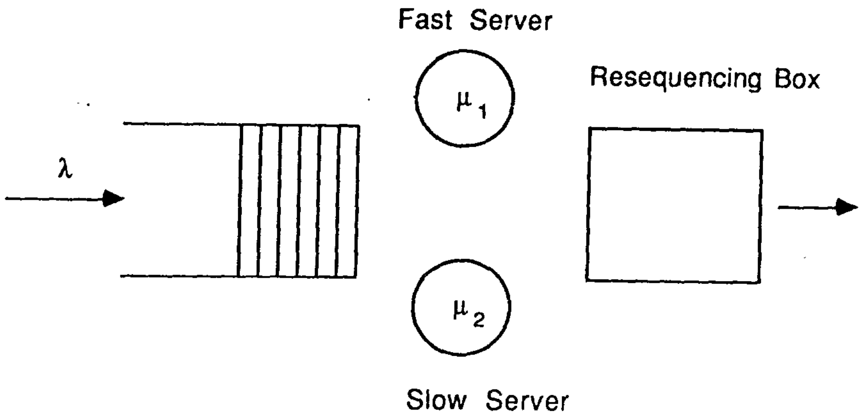

Fig 4.2.1 The Resequencing System

4.2 A Markovian State Space Description

Consider a $M/M/2/B$ queue with arrival rate λ, and service rates of magnitude μ_1 and μ_2 for servers one and two, respectively (Fig 4.2.1). Assume that $\mu_1 \geq \mu_2$ so that server one (1) and server two (2) can be called the fast and slow servers, respectively. Pose

$n =$ number of customers in the main queue buffer.

$e_1 = 1$ (resp. 0) if the faster server is busy (resp. idle).

$e_2 = 1$ (resp. 0) if the slower server is busy (resp. idle).

$m =$ number of customers in the resequencing buffer.

The variables (n, e_1, e_2, m) do not constitute a Markovian description of the system, since there is no way to take into account the effect on m by a service completion at either server. Due to the synchronization constraint on the output customer stream, we need a state variable which captures this effect. A clever way of defining this state which was first given by Luke Lien [49], is now presented. The additional information needed to get a Markovian state space description is the specification of which of the two customers presently in service, started receiving service earlier (Fig 4.2.2). This is exactly what the fifth state variable, denoted

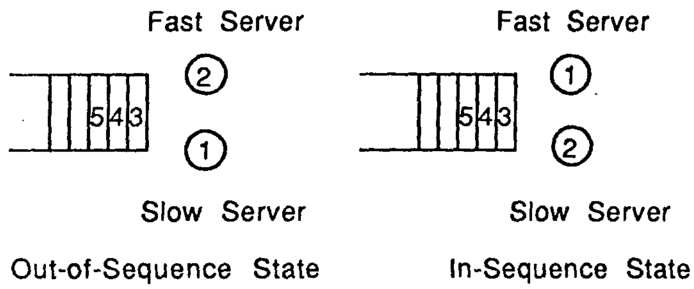

Fig 4.2.2 O-State and I-State

by Z, specifies with

$Z = I$ if the fast server (1) is serving the customer which entered the system earlier. We shall refer to this as being an *in-sequence* state.

$Z = O$ if the slow server (2) is serving the customer which entered the system earlier. We shall refer to this as being an *out-of-sequence* state.

When there is a single customer in the system, we shall adopt the same notation with the interpretation that $Z = I$ if the customer is with the fast server and $Z = O$ if the customer is with the slow server.

The reader will readily check that (n, e_1, e_2, m, Z) provides a complete Markovian state space description of the system. The state variables (n, e_1, e_2, m, Z) belong to the space

$$E = \{0\} \cup \mathbb{N} \times \{0,1\} \times \{0,1\} \times \mathbb{N} \times \{I, O\}$$

where $\{0\}$ is the 'empty' state.

If the system is in a in-sequence state ($Z = I$), then a departure from server 2 leads to an increase in the number of customers in the resequencing box by one ($m \to m + 1$), since the customer who arrived earlier is being served by server 1.

On the other hand, a departure from server 1 empties all the customers in the resequencing buffer ($m \rightarrow 0$), and changes the state to an out-of-sequence state (if there is a customer in service in server 2). By a similar reasoning, if the system is in an out-of-sequence state ($Z = O$), a departure from server 1 leads to an increase in the number of customers in the resequencing box ($m \rightarrow m + 1$), while a departure from server 2 empties the resequencing box ($m \rightarrow 0$).

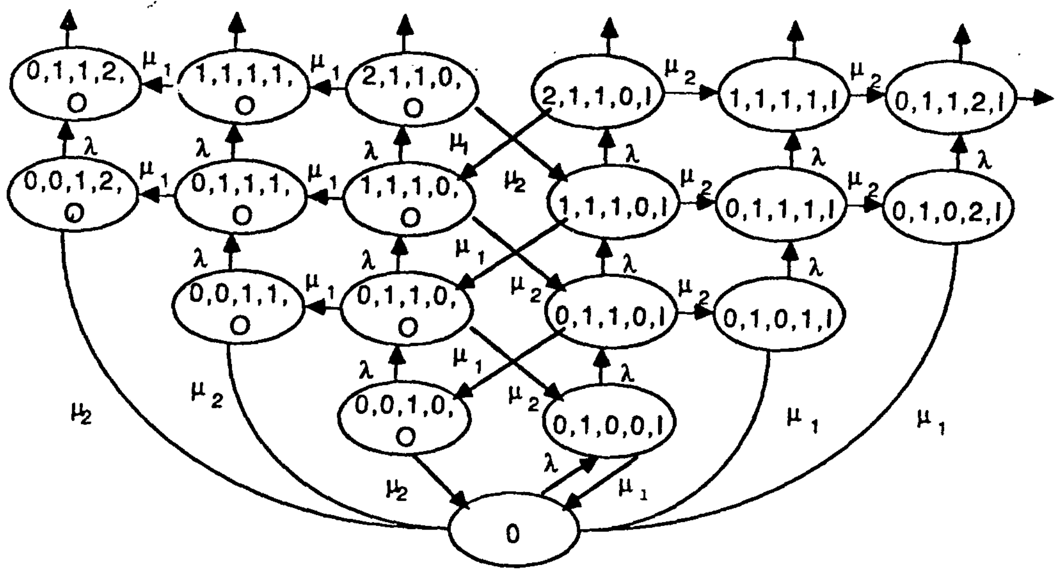

Fig 4.3.1. The State Space

4.3 The State Space Equations

In this section we proceed to write down the equations for the steady state probabilities for the Markov Chain associated with the $M/M/2/B$ queue with resequencing. This Markov chain is partially illustrated in Fig 4.3.1., however in the interests of clarity, some of the transistions have been left out of the figure. The complete state equations are provided below.

1. The equilibrium equation at the origin.

$$\lambda P(0) = \mu_1 \sum_{j=0}^{\infty} P(0,1,0,j,I) + \mu_2 \sum_{j=0}^{\infty} P(0,0,1,j,O) \tag{4.1}$$

2. The equlibrium equations for the states for which $Z = I$.

(a) For $0 < i < B, j > 0, e_1 = 1, e_2 = 1$.

$$(\lambda + \mu_1 + \mu_2)P(i,1,1,j,I) = \mu_2 P(i+1,1,1,j-1,I) + \lambda P(i-1,1,1,j,I) \quad (4.2a)$$

(b) For $i = B, j \geq 0, e_1 = 1, e_2 = 1$.

$$(\mu_1 + \mu_2)P(B,1,1,j,I) = \lambda P(B-1,1,1,j,I) \quad (4.2b)$$

(c) For $i = 0, j > 0, e_1 = 1, e_2 = 1$.

$$(\lambda + \mu_1 + \mu_2)P(0,1,1,j,I) = \mu_2 P(1,1,1,j-1,I) + \lambda P(0,1,0,j,I) \quad (4.2c)$$

(d) For $i = 0, j > 0, e_1 = 1, e_2 = 0$.

$$(\lambda + \mu_1)P(0,1,0,j,I) = \mu_2 P(0,1,1,j-1,I) \quad (4.2d)$$

(e) For $0 < i < B, j = 0, e_1 = 1, e_2 = 1$.

$$(\lambda + \mu_1 + \mu_2)P(i,1,1,0,I) = \mu_2 \sum_{j=0}^{\infty} P(i+1,1,1,j,O) + \lambda P(i-1,1,1,0,I) \quad (4.2e)$$

(f) For $i = 0, j = 0, e_1 = 1, e_2 = 0$.

$$(\lambda + \mu_1 + mu_2)P(0,1,1,0,I) = \mu_2 \sum_{j=0}^{\infty} P(1,1,1,j,O) + \lambda P(0,1,0,0,I) \quad (4.2f)$$

(g) For $i = 0, j = 0, e_1 = 1, e_2 = 0$.

$$(\lambda + \mu_1)P(0,1,0,0,I) = \mu_2 \sum_{j=0}^{\infty} P(0,1,1,j,O) + \lambda P(0,0,0.0) \quad (4.2g)$$

2. The equilibrium equations for the states for which $Z = O$

(a) For $0 < i < B, j > 0, e_1 = 1, e_2 = 1$.

$$(\lambda + \mu_1 + \mu_2)P(i,1,1,j,O) = \mu_1 P(i+1,1,1,j-1,O) + \lambda P(i-1,1,1,j,O) \quad (4.3a)$$

(b) For $i = B, j \geq 0, e_1 = 1, e_2 = 1$.

$$(\mu_1 + \mu_2)P(B, 1, 1, j, O) = \lambda P(B-1, 1, 1, j, O) \tag{4.3b}$$

(c) For $i = 0, j > 0, e_1 = 1, e_2 = 1$.

$$(\lambda + \mu_1 + \mu_2)P(0, 1, 1, j, O) = \mu_1 P(1, 1, 1, j-1, O) + \lambda P(0, 0, 1, j, O) \tag{4.3c}$$

(d) For $i = 0, j > 0, e_1 = 0, e_2 = 1$.

$$(\lambda + \mu_2)P(0, 0, 1, j, O) = \mu_1 P(0, 1, 1, j-1, O) \tag{4.3d}$$

(e) For $0 < i < B, j = 0, e_1 = 1, e_2 = 1$.

$$(\lambda + \mu_1 + \mu_2)P(i, 1, 1, 0, O) = \mu_1 \sum_{j=0}^{\infty} P(i+1, 1, 1, j, O) + \lambda P(i-1, 1, 1, 0, O) \tag{4.3e}$$

(f) For $i = 0, j = 0, e_1 = 1, e_2 = 1$.

$$(\lambda + \mu_1 + \mu_2)P(0, 1, 1, 0, O) = \mu_1 \sum_{j=0}^{\infty} P(1, 1, 1, j, I) + \lambda P(0, 0, 1, 0, O) \tag{4.3f}$$

(g) For $i = 0, j = 0, e_1 = 0, e_2 = 1$.

$$(\lambda + \mu_2)P(0, 0, 1, 0, O) = \mu_1 \sum_{j=0}^{\infty} P(0, 1, 1, j, I) \tag{4.3g}$$

4.4 The Case $B = 0$

Explicit closed form expressions can be obtained for the buffer occupation probabilities for the special case when $B = 0$. A customer who arrives when both the servers are busy is discarded. Because of the resequencing constraint, customers leave the system in the same order in which they started service. We assume that the resequencing box has unlimited buffer space.

Note that all the results given below can be recovered from the more general discussion of Section 4.5. We nevertheless go through the calculations because the case $B = 0$ is of interest in its own right and the equations being much simpler than for the general case, it serves an illustrative purpose.

The equations to be solved are now stated below. Since $n = 0$ everywhere, it is omitted from the notation. The equations (4.1)-(4.3) now become,

$$\lambda P(0) = \mu_1 \sum_{j=0}^{\infty} P(1,0,j,I) + \mu_2 \sum_{j=0}^{\infty} P(0,1,j,I) \qquad j \geq 0 \tag{4.4}$$

$$(\lambda + \mu_1)P(1,0,j,I) = \mu_2 P(1,1,j-1,I) \qquad j \geq 1 \tag{4.5a}$$

$$(\lambda + \mu_2)P(0,1,j,O) = \mu_1 P(1,1,j-1,O) \qquad j \geq 1 \tag{4.5b}$$

$$(\mu_1 + \mu_2)P(1,1,j,I) = \lambda P(1,0,j,I) \qquad j \geq 0 \tag{4.6a}$$

$$(\mu_1 + \mu_2)P(1,1,j,O) = \lambda P(0,1,j,O) \qquad j \geq 0 \tag{4.6b}$$

$$(\lambda + \mu_1)P(1,0,0,I) = \lambda P(0) + \mu_2 \sum_{j=0}^{\infty} P(1,1,j,O) \tag{4.7a}$$

$$(\lambda + \mu_2)P(0,1,0,O) = \mu_1 \sum_{j=0}^{\infty} P(1,1,j,I) \tag{4.7b}$$

We now proceed to solve these equations. From (4.5a-b) and (4.6a-b) it is easy to see that the equations

$$P(1,0,j,I) = \left(\frac{\mu_2}{\lambda + \mu_1}\right)^j \left(\frac{\lambda}{\mu_1 + \mu_2}\right)^j P(1,0,0,I) \qquad j \geq 0 \tag{4.8}$$

$$P(1,1,j,I) = \left(\frac{\mu_2}{\lambda + \mu_1}\right)^j \left(\frac{\lambda}{\mu_1 + \mu_2}\right)^{j+1} P(1,0,0,I) \qquad j \geq 0 \tag{4.9}$$

$$P(0,1,j,O) = (\frac{\mu_1}{\lambda+\mu_2})^j (\frac{\lambda}{\mu_1+\mu_2})^j P(0,1,0,O) \qquad j \geq 0 \tag{4.10}$$

$$P(1,1,j,O) = (\frac{\mu_1}{\lambda+\mu_2})^j (\frac{\lambda}{\mu_1+\mu_2})^{j+1} P(0,1,0,O) \qquad j \geq 0 \tag{4.11}$$

are satisfied.

Substituting (4.11) into (4.7a), we obtain

$$(\lambda+\mu_1)P(1,0,0,I) = \lambda P(0) + \mu_2 \sum_{j=0}^{\infty} (\frac{\mu_1}{\lambda+\mu_2})^j (\frac{\lambda}{\mu_1+\mu_2})^{j+1} P(0,1,0,O) \tag{4.12}$$

Pose

$$\begin{aligned}\sigma_1 &= \sum_{j=0}^{\infty} (\frac{\mu_1}{\lambda+\mu_2})^j (\frac{\lambda}{\mu_1+\mu_2})^{j+1} \\ &= \frac{\lambda(\lambda+\mu_2)}{\mu_2(\lambda+\mu_1+\mu_2)}\end{aligned}$$

with σ_1 always finite since

$$\frac{\mu_1}{\lambda+\mu_2} \frac{\lambda}{\mu_1+\mu_2} < 1$$

Hence (4.12) can be rewritten as

$$(\lambda+\mu_1)P(1,0,0,I) = \lambda P(0) + \sigma_1 \mu_2 P(0,1,0,O) \tag{4.13}$$

Substituting (4.9) into (4.7b), we also obtain

$$(\lambda+\mu_2)P(0,1,0,O) = \mu_1 \sum_{j=0}^{\infty} (\frac{\mu_2}{\lambda+\mu_1})^j (\frac{\lambda}{\mu_1+\mu_2})^{j+1} P(1,0,0,I) \tag{4.14}$$

Pose

$$\begin{aligned}\sigma_2 &= \sum_{j=0}^{\infty} (\frac{\mu_2}{\lambda+\mu_1})^j (\frac{\lambda}{\mu_1+\mu_2})^{j+1} \\ &= \frac{\lambda(\lambda+\mu_1)}{\mu_1(\lambda+\mu_1+\mu_2)}\end{aligned}$$

with σ_2 obviously finite since

$$\frac{\mu_2}{\lambda+\mu_1}\frac{\lambda}{\mu_1+\mu_2} < 1.$$

Hence (4.14) can be rewritten as

$$(\lambda+\mu_2)P(0,1,0,O) = \sigma_2\mu_1 P(1,0,0,I) \tag{4.15}$$

The relations (4.13) and (4.15) provide us with two equations for the unknown values of $P(0), P(1,0,0,I)$ and $P(0,1,0,O)$. In order to get a third equation, we use the fact that the sum of all the probabilities should be one, i.e.,

$$P(0) + \sum_{j=0}^{\infty}(P(1,0,j,I) + P(1,1,j,I) + P(0,1,j,O) + P(1,1,j,O)) = 1 \tag{4.16}$$

Substituting from (4.8)-(4.11), into (4.16) we obtain

$$P(0) + (\sigma_2+\sigma_3)P(1,0,0,I) + (\sigma_1+\sigma_4)P(0,1,0,O) = 1 \tag{4.17}$$

where σ_1 and σ_2 are as defined earlier and σ_3 and σ_4 are given by

$$\begin{aligned}\sigma_3 &= \sum_{j=0}^{\infty}\left(\frac{\mu_2}{\lambda+\mu_1}\right)^j\left(\frac{\lambda}{\mu_1+\mu_2}\right)^j \\ &= \frac{(\lambda+\mu_1)(\mu_1+\mu_2)}{\mu_1(\lambda+\mu_1+\mu_2)}\end{aligned}$$

and

$$\begin{aligned}\sigma_4 &= \sum_{j=0}^{\infty}\left(\frac{\mu_1}{\lambda+\mu_2}\right)^j\left(\frac{\lambda}{\mu_1+\mu_2}\right)^j \\ &= \frac{(\lambda+\mu_2)(\mu_1+\mu_2)}{\mu_2(\lambda+\mu_1+\mu_2)}.\end{aligned}$$

The values of $P(0), P(1,0,0,I)$ and $P(0,1,0,O)$ can now be obtained very easily by solving the system of linear equations

$$\begin{pmatrix} \lambda & -(\lambda+\mu_1) & \sigma_1\mu_2 \\ 0 & \sigma_2\mu_1 & -(\lambda+\mu_2) \\ 1 & (\sigma_2+\sigma_3) & (\sigma_1+\sigma_4) \end{pmatrix} \begin{pmatrix} P(0) \\ P(1,0,0,I) \\ P(0,1,0,O) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \tag{4.18}$$

With

$$\psi := \frac{(\lambda+\mu_1)(\lambda+\mu_2)}{\sigma_2\mu_1\lambda} - \frac{\sigma_1\mu_2}{\lambda} + \frac{(\sigma_2+\sigma_3)(\lambda+\mu_2)}{\sigma_2\mu_1} + \sigma_1 + \sigma_4 \tag{4.19}$$

routine yet tedious calculations give

$$P(0) = \frac{(\lambda+\mu_1)(\lambda+\mu_2)}{\sigma_2\mu_1\lambda\psi} - \frac{\sigma_1\mu_2}{\lambda\psi} \tag{4.20}$$

$$P(1,0,0,I) = \frac{(\lambda+\mu_2)}{\sigma_2\mu_1\psi} \tag{4.21}$$

$$P(0,1,0,O) = \frac{1}{\psi} \tag{4.22}$$

We can use (4.8)-(4.11) to obtain the values the other steady state probabilities.

1) The probability that there are j customers in the resequencing buffer and the customer who arrived earlier is being served by the fast server.

$$P(j,I) = (\frac{\mu_2}{\lambda+\mu_1})^j(\frac{\lambda}{\mu_1+\mu_2})^j(\frac{\lambda+\mu_1+\mu_2}{\mu_1+\mu_2})\frac{\lambda+\mu_2}{\sigma_2\mu_1\psi}, \quad j \geq 0. \tag{4.23}$$

2) The probability that there are j customers in the resequencing buffer and the customer who has arrived earlier is being served by the slow server.

$$P(j,O) = (\frac{\mu_1}{\lambda+\mu_2})^j(\frac{\lambda}{\mu_1+\mu_2})^j\frac{\lambda+\mu_1+\mu_2}{\mu_1+\mu_2}\frac{1}{\psi}, \qquad j \geq 0. \tag{4.24}$$

3) The probability that there are j customers in the resequencing buffer.

$$P(j) = (\frac{\lambda}{\mu_1+\mu_2})^j\frac{\lambda+\mu_1+\mu_2}{\mu_1+\mu_2}[(\frac{\mu_2}{\lambda+\mu_1})^j\frac{\lambda+\mu_2}{\sigma_2\mu_1\psi} + (\frac{\mu_1}{\lambda+\mu_2})^j]\frac{1}{\psi}, \qquad j \geq 0. \tag{4.25}$$

4.5 The General Case

In the present section we present a technique for calculating the exact values of the buffer occupation probabilities in the $M/M/2/B$ queue with resequencing. Note that equations (4.8)-(4.11) in the last section indicate a geometric structure for the buffer occupation probabilities when B=0. We carry that insight to its logical conclusion by showing that in the general case, the buffer occupation probabilities have a *matrix-geometric* structure.

We proceed as follows. The states in the Markov chain are numbered appropriately so that the corresponding infinitesmal generator matrix Q is seen to have matrix-geometric structure. In fact the structure coincides with the modified matrix associated with complex boundary behaviour which is identified in Neuts, P. 24 [56]. Once this is done, the probability vector can be written down using standard techniques.

As the first step we partition the state probability vector into the vectors $(P(0), \pi_0, \pi_1, \ldots)$, where $P(0)$ is the probability of the zero state as before and

$$\pi_j = (P(0,0,1,j,O), P(0,1,1,j,O), \ldots, P(B,1,1,j,O), P(B,1,1,j,I), \ldots \\ \ldots, P(0,1,1,j,I), P(0,1,0,j,I)) \tag{4.26}$$

for all $j \geq 0$. Hence π_j is a $(1 \times 2(B+2))$ row vector which contains the probabilities of all states that have j customers in the resequencing buffer. Using this partition of the state probability vector, we can write the infitesmal generator matrix Q in the block partition form

$$Q = \begin{pmatrix} D0 & C0 & 0 & 0 & 0 & 0 & \ldots \\ D1 & C1 & A0 & 0 & 0 & 0 & \ldots \\ D1 & C2 & A1 & A0 & 0 & 0 & \ldots \\ D1 & C2 & 0 & A1 & A0 & 0 & \ldots \\ D1 & C2 & 0 & 0 & A1 & A0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{pmatrix} \tag{4.27}$$

In (4.27), $D0 = -\lambda$ and the other matrices are defined below, with the convention $\gamma = (\lambda + \mu_1 + \mu_2)$.

$$C0 = (0,0,\ldots,0,\lambda)^{1\times 2(B+2)}$$

$$D1^T = (\mu_2,0,\ldots,0,\mu_1)^{1\times 2(B+2)}$$

$$A0 = \begin{pmatrix} A0_{11} & 0 \\ 0 & A0_{22} \end{pmatrix}$$

where

$$A0_{11} = \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ \mu_1 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & \mu_1 & 0 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & \mu_1 & 0 & 0 \\ 0 & 0 & 0 & \ldots & 0 & \mu_1 & 0 \end{pmatrix}^{(B+2)\times(B+2)}$$

and

$$A0_{22} = \begin{pmatrix} 0 & \mu_2 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & \mu_2 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 0 & \mu_2 & 0 \\ 0 & 0 & 0 & \ldots & 0 & 0 & \mu_2 \\ 0 & 0 & 0 & \ldots & 0 & 0 & 0 \end{pmatrix}^{(B+2)\times(B+2)}$$

$$A1 = \begin{pmatrix} A1_{11} & 0 \\ 0 & A1_{22} \end{pmatrix}$$

where

$$A1_{11} = \begin{pmatrix} -(\lambda+\mu_2) & \lambda & 0 & \ldots & 0 & 0 & 0 \\ 0 & -\gamma & \lambda & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 0 & -\gamma & \lambda \\ 0 & 0 & 0 & \ldots & 0 & 0 & -(\mu_1+\mu_2) \end{pmatrix}^{(B+2)\times(B+2)}$$

and

$$A1_{22} = \begin{pmatrix} -(\mu_1+\mu_2) & 0 & 0 & \ldots & 0 & 0 & 0 \\ \lambda & -\gamma & 0 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & \lambda & -\gamma & 0 \\ 0 & 0 & 0 & \ldots & 0 & \lambda & -(\lambda+\mu_1) \end{pmatrix}^{(B+2)\times(B+2)}$$

$$C2 = \begin{pmatrix} 0 & C2_{12} \\ C2_{21} & 0 \end{pmatrix}$$

where

$$C2_{12} = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & \mu_2 \\ 0 & 0 & 0 & \dots & 0 & \mu_2 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \mu_2 & \dots & 0 & 0 & 0 \\ 0 & \mu_2 & 0 & \dots & 0 & 0 & 0 \end{pmatrix}^{(B+2)\times(B+2)}$$

and

$$C2_{21} = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & \mu_1 & 0 \\ 0 & 0 & 0 & \dots & \mu_1 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \mu_1 & 0 & \dots & 0 & 0 & 0 \\ \mu_1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 \end{pmatrix}^{(B+2)\times(B+2)}$$

$$C1 = \begin{pmatrix} C1_{11} & C1_{12} \\ C1_{21} & C1_{22} \end{pmatrix}$$

where

$$C1_{11} = A1_{11}, C1_{12} = C2_{12}, C1_{21} = C2_{21}, C2_{22} = A1_{22}$$

Let e be a $(2(B+2) \times 1)$ column vector with all its components equal to one. Since Q is an infinitesmal generator matrix, its rows should sum upto zero,i.e.,

$$\begin{aligned} D0 + C0e &= 0 \\ D1 + C1e + A0e &= 0 \\ D1 + C2e + A1e + A0e &= 0 \end{aligned} \qquad (4.28)$$

We now proceed with the task of solving the equations

$$\pi Q = 0, \qquad \pi e = 1 \qquad (4.29)$$

which can be rewritten as

$$P(0)D0 + D1 \sum_{i=0}^{\infty} \pi_i = 0 \tag{4.30}$$

$$P(0)C0 + \pi_0 C1 + C2 \sum_{i=1}^{\infty} \pi_i = 0 \tag{4.31}$$

$$\pi_i A0 + \pi_{i+1} A1 = 0 \qquad i \geq 0 \tag{4.32}$$

$$P(0) + \sum_{i=0}^{\infty} \pi_i = 1 \tag{4.33}$$

Before we can solve (4.30)-(4.33) we need the following Lemma.

Lemma 4.5.1. *The following statements hold true, namely*

(1) The matrix $A1$ *is nonsingular.*

(2) If

$$R = -A0(A1^{-1}) \tag{4.34}$$

then the eigenvalue $\lambda(R)$ *of* R *with largest modulus satisfies the condition,*

$$\lambda(R) < 1. \tag{4.35}$$

(3) The matrix $B(R)$ *defined by*

$$B(R) = \begin{pmatrix} D0 & C0 \\ (\sum_{i=0}^{\infty} R^i)D1 & C1 + (\sum_{i=1}^{\infty} R^i)C2 \end{pmatrix} \tag{4.36}$$

is an infinitesmal generator matrix.

Proof. (1) The nonsingularity of $A1$ can be proved very easily as follows. If the row vector $u = (u_1, u_2)$ is in the (left) null space of $A1$, then

$$uA1 = 0 \tag{4.37}$$

and this implies that

$$(\lambda + \mu_2)u_1 = 0 \qquad \text{and} \qquad (\lambda + \mu_1)u_2 = 0 \tag{4.38}$$

whence $u_1 = u_2 = 0$, i.e., $A1$ is nonsingular.

(2) Since $A1$ is nonsingular, the matrix R is well defined. Let u be the left eigenvector of R corresponding to $\lambda(R)$, i.e.,

$$uR = \lambda(R)u \tag{4.39}$$

or equivalently,

$$-uA0A1^{-1} = \lambda(R)u. \tag{4.40}$$

Therefore

$$-uA0 = \lambda(R)uA1 \tag{4.41}$$

and postmultiplying by e on both sides of (4.33) leads to

$$\lambda(R) = -\frac{uA0e}{uA1e}. \tag{4.42}$$

It is easy to see that

$$uA0e = \mu_1(\sum_{i=2}^{B+2} u_i) + \mu_2(\sum_{i=B+3}^{2B+3} u_i) \tag{4.43}$$

$$-uA1e = \mu_1(\sum_{i=2}^{2B+4} u_i) + \mu_2(\sum_{i=1}^{2B+3} u_i) \tag{4.44}$$

The conclusion $\lambda(R) < 1$ now follows from (4.43)-(4.44).

(3) Since $\lambda(R) < 1$, it follows that $\sum_{i=0}^{\infty} R^i = (I - R)^{-1}$ is well defined. To prove that $B(R)$ is an infinitesmal generator, first note that $D0 + C0e = 0$ by (4.28). Hence it suffices to verify that

$$\sum_{i=1}^{\infty} R^i(D1 + C2e) + D1 + C1e = 0. \tag{4.45}$$

This is easily done by direct calculations whose details are left to the interested reader. ∎

We can now state the main result in this section.

Theorem 4.5.1. *The solution to (4.25) is given by the vector* $\pi = (P(0), \pi_0, \pi_1, \ldots)$ *where*

$$\pi_i = \pi_0 R^i, \qquad i \geq 0 \tag{4.46}$$

with R defined by (4.34) and $(P(0), \pi_0)$ *solves the eqn*

$$(\, P(0) \quad \pi_0 \,)\, B(R) = 0 \tag{4.47}$$

subject to the normalization condition

$$P(0) + \pi_0 (I - R)^{-1} e = 1 \tag{4.48}$$

Proof. By Lemma 4.5.1, A1 in nonsingular, so that (4.46) follows directly from (4.32). Also (4.47) follows from (4.30)-(4.31) after substituting for $\{\pi_i, i \geq 1\}$ in terms of π_0 via (4.46). ∎

The distribution of the number of customers in the resequencing buffer can be recovered from Theorem 4.5.1. From the definition of π_i given in (4.26), the probability q_j of finding j customers in the resequencing buffer is simply the sum of the probabilities in π_j, therefore,

$$q_j = \begin{cases} \pi_j e = \pi_0 R^j e & \text{if} \quad j \geq 1; \\ 1 - \pi_0 R (I - R)^{-1} e & \text{if} \quad j = 0. \end{cases} \tag{4.49}$$

The average number of customers in the resequencing buffer is then given by

$$\begin{aligned} \overline{N} &= \sum_{j=1}^{\infty} j\, q_j \\ &= \sum_{j=1}^{\infty} j\, \pi_0 R^j e \\ &= \pi_0 \sum_{j=1}^{\infty} j\, R^j e \\ &= \pi_0 R (I - R)^{-2} e \end{aligned} \tag{4.50}$$

4.6 Bulk Departure Size Distribution of the $M/M/2$ Queue with Resequencing

We give formulae for the bulk departure size distribution for the case when $B = \infty$ and $\mu_1 = \mu_2 = \mu$. The assumption of equal service rate for the two servers is made so that the bulk departure has equal probability of being trigered by service completion at either server. In the case of unequal service rates, the bulk departure clearly has different probabilities of being caused by a departure from either the fast or the slow server. Since we have been unable to estimate this probability, the assumption of equal servive rates is necessary. In the rest of the section, a customer responsible for trigerring the bulk departure, is referred to as a *star customer.*

When the number of servers is two, the calculation of the distribution of the bulk departure process from the resequencing buffer is simplified by the fact that the star customer clears the resequencing buffer of all customers, when he departs. In other words the number of customers in the departing bulkcoincides with the number of customers that the star customer finds in the resequencing buffer (plus the star customer himself). But note that the number of customers that the star customer finds in the resequencing buffer is equal to the number of customers that overtook him by going through the other server while the star customer was being served. Hence all that we have to do is to find the distribution of the number of occurences of the point process corresponding to service completion at a server, while a star customer is in service at the other server. There is one case when this can be done easily, i.e. when the buffer size is infinite, since in this case the point process of interest happens to be Poisson. In the case of finite buffer size, the point process far less from being Poisson, is not even a renewal process [18], thus rendering the calculation of the bulk departure distribution more difficult.

- The case $B = \infty$

Consider a $M/M/2$ queue with resequencing at equilibrium. Assume that the star customer is in service at the first server. It is easy to see that the rest of the queue behaves like a $M/M/1$ queue with service rate μ and arrival rate λ. By Burke's Theorem, the output process from the queue is Poisson with rate λ.

Hence we have to find the number of points of a Poisson process with rate λ, in an interval whose length exponentially distributed with parameter μ. Let the length of the service interval of the star customer be T. If $N(t)$ is the number of points of the Poisson process in an interval of length t, then it follows that

$$\Pr(N(T) = k \mid T = t) = \exp(-\lambda t)\frac{(\lambda t)^k}{k!} \qquad k = 0, 1 \ldots \quad (4.51)$$

Using the Law of Total Probabilitie, we obtain

$$\begin{aligned}\Pr(N(T) = k) &= \int_0^\infty \exp(-\lambda t)\frac{(\lambda t)^k}{k!}\mu\exp(\mu t)\,dt \\ &= \frac{\mu}{(\lambda+\mu)}(\frac{\lambda}{\lambda+mu})^k\end{aligned} \qquad k = 0, 1 \ldots \quad (4.52)$$

(4.52) gives the distribution of the number of the number of customers which overtake the star customer while he is in service in the first server. The distribution of the number of customers which overtake the the star customer while he is in service in the slower server is given by exactly the same expression due to the asssumption of equal service rates.

Define b_k=P(the probability that the departing bulk consists of k customers). From the above discussion it is clear that

$$b_{k+1} = \frac{\mu}{(\lambda+\mu)}(\frac{\lambda}{\lambda+\mu})^k \qquad k = 0, 1 \ldots \quad (4.53)$$

CHAPTER V

THE MAIN RESULTS

5.1 Introduction

In this chapter we deal with a simple control problem that arises in a system with synchronization constraints. The queueing system under consideration is the $M/M/2$ queue with resequencing analysed Chapter 4 where the servers had unequal service rates. The new feature in the system is a scheduler which assigns customers from the main buffer to the servers so as to minimize the average end-to-end delay of the customers which enter the system. This problem is similar to the one that was analysed by Kumar and Lin [41] except for the fact that now the customers are constrained to leave the system in the same order in which they entered it.

If we regard the two servers as communication links and the customers as the message packets, then this problem is of immediate practical importance. Similar conditions arise in a flexible manufacturing system also.

Our main result concerns the structure of the optimal policy which assigns customers to the two servers so as to minimize the end-to-end delay of the customers, including the resequencing delay. It is shown that the faster server should be kept busy whenever possible and the decision to send a customer to the slower server is independent of the number of customers in the resequencing buffer, being of the threshold type in the number of customers in the main queue buffer. The policy is thus of the type identified by Kumar and Lin but the threshold will obviously be different in the two cases. Though we have not been able to derive an explicit formula for the optimal threshold, it is intuitively evident that it should be no smaller than the threshold for the case treated by Kumar and Lin since resequencing makes the use of the slower server more expensive. We will consider this issue in greater detail later.

An attractive aspect of the optimal policy is that the scheduler does not have to care about the number of customers in the resequencing buffer. In the communications link example, this means that we can optimally assign messages to the two links without keeping account of the number of messages at the receiver which are out of sequence. Keeping this record would have entailed an additional communications cost.

In the present chapter we give a description of the problem and state the equations that must be satisfied in order to identify the optimal control of the two servers, while in the next chapter we provide an inductive proof of these equations using the standard value iteration technique of dynamic programming.

This chapter is organized as follows. The model is introduced in Section 5.2. In Section 5.3, we give a discrete time formulation of the problem, and an explicit description of the control actions and events is provided in section 5.4. The dynamic programming equation is then formulated in Section 5.5 The optimal control for the faster server is identified in Section 5.6., while the optimal control of the slower server is identified in Section 5.7.

5.2 The Queueing Model

The system under investigation is a $M/M/2$ queue with heterogenous service rates and resequencing (Fig 4.2.1). The input process is Poisson with rate λ. The two service time distributions are also exponential with rates μ_1 and μ_2 at server 1 and server 2, respectively. We assume that $\mu_1 > \mu_2$ so that server 1 and server 2 can be called the fast and slow servers respectively. A state space representation of the system is provided by the quintuple $x = (x_0, x_1, x_2, x_3, Z)$ where

Number of customers in the buffer of the $M/M/2$ queue.

1 (resp. 0) if the fast server is busy (resp. idle).

1 (resp. 0) if the slow server is busy (resp. idle).

number of customers in the resequencing buffer.

I (resp. O) if the customer being served by server 1 (resp. server 2), arrived earlier than the one being served by server 2. (resp. server 1).

When there is a single customer being served by either one of the two servers, we shall adopt the same notation with the interpretation that $Z = I$ if the customers is with server 1 and $Z = O$ if the customer is with server 2.

Also note that the model does not support jockeying of customers between the two servers so that once a customers commences service at some server, it remains there for the duration of its service.

5.3 Description of Problem

Our aim is to find the optimal policy which assigns customers to the two servers in the queueing model described in Section 5.2, so as to minimize their average end-to-end delay. To that end, define the cost incurred per unit time at time t, with the system in state $x(t)$ as $c(t) = x_0(t) + x_1(t) + x_2(t) + x_3(t)$. This is seen to be linear in the number of customers in the system.

A policy γ is any rule which at $t \geq 0$ decides on the basis of $\{x(s), s \leq t\}$, whether to send a queued message to the idle servers or not. Since we consider only the discounted cost case in this chapter, let $\alpha \geq 0$ be the interest rate used for discounting the future cost, i.e., the present value of cost $c(t)$ incurred at time t is $c(t)\exp(-\alpha t)$ so that the cost incurred by a policy γ over the interval $[0, \infty)$ with initial state x is given by

$$J(x, \gamma) = E_x^\gamma \int_0^\infty \exp^{-\alpha t} \sum_{i=0}^{3} x_i(t) dt. \tag{5.1}$$

A policy π is optimal if $J(x, \pi) = \inf_\gamma J(x, \gamma)$ and it is well known that π is Markov and stationary.

5.4 Conversion into a Discrete Time Problem.

It is useful to give a discrete time formulation of the problem in order to facilitate the use of the inductive approach in the dynamic programming argument. It is shown by Lippman [46] and Serfozo [68], that if a controlled Markov process M_1, has transition rates uniformly bounded by some some constant c, then another controlled Markov process M_2, can be constructed which is probabilistically equivalent to it i.e. they have the same infinitesmal generator matrix Q. Morever M_2 is such that the time between all transitions are exponentially distributed with the parameter c, independent of the state of the process. Hence a complete description of M_2 can be given by means of c and a discrete time Markov chain M_3, imbedded at the times of the state transitions of M_2. It can be easily shown that the transition probability matrix of M_3 is given by $I + \frac{Q}{c}$. It is shown in [46], that if a stationary policy is employed for the infinite horizon problem, then both the discrete and continous formulations are equivalent, as are their functional equations. Hence we can equivalently work with the discrete time formulation of the problem without losing anything as far as the identification of the optimal policy is concerned.

Since in our problem the transition rates of the original continous time Markov process are bounded by $(\lambda + \mu_1 + \mu_2)$, we can apply the above mentioned results and transform it into a discrete time process. Note that the arrival service rates in the discrete time problem are different than in the original continous time problem, in fact their new value depends on the parameter c. We choose c in such a way that

$$\lambda + \mu_1 + \mu_2 = 1$$

in the discrete time system.

5.5 Description of Discrete Time Problem

In this section we provide a detailed description of the events and control actions in the discrete time problem.

The system state $x = (x_0, x_1, x_2, x_3, Z)$ belongs to the state space

$$\begin{aligned} E = \{0\} &\cup I\!N \times \{0,0,0,Z\} \\ &\cup I\!N \times \{1,0\} \times I\!N \times \{I\} \\ &\cup I\!N \times \{0,1\} \times I\!N \times \{O\} \\ &\cup I\!N \times \{1,1\} \times I\!N \times \{I,O\} \end{aligned} \tag{5.2}$$

where $\{0\}$ represents the 'empty' state.

Define the operators A, D_1, D_2 $E \to E$ as follows:

$$A(x_0, x_1, x_2, x_3, Z) = (x_0 + 1, x_1, x_2, x_3, Z) \tag{5.3}$$

$$D_1(x_0, 1, x_2, x_3, I) = (x_0, 0, x_2, 0, O) \tag{5.4a}$$

$$D_1(x_0, 0, x_2, x_3, O) = (x_0, 0, x_2, x_3, O) \tag{5.4b}$$

$$D_1(x_0, 1, x_2, x_3, O) = (x_0, 0, x_2, x_3 + 1, O) \tag{5.4c}$$

$$D_2(x_0, x_1, 1, x_3, O) = (x_0, x_1, 0, 0, I) \tag{5.5a}$$

$$D_2(x_0, x_1, 0, x_3, I) = (x_0, x_1, 0, x_3, I) \tag{5.5b}$$

$$D_2(x_0, x_1, 1, x_3, I) = (x_0, x_1, 0, x_3 + 1, I) \tag{5.5c}$$

It is plain that A is the arrival operator and that $D_i, i = 1,2$ is the departure operator from server i. Depending on whether $Z = I$ or O, a departure from server 1 either adds to the number in the resequencing buffer by one, or clears it of all customers. The same behaviour is exhibited by a departure from server 2. The changes in the Z component are illustrated by the following example: When $x_1 = x_2 = 1$ and $Z = I$, then a departure from server 1 changes I to O, since following the departure, server 2 is left serving the customer who started service earlier. Note that a departure of a dummy customer does not change the state of the system.

We now define the control operators for assigning customers from the buffer to the servers.

- The hold operator P_h is first defined by

$$P_h(x_0, x_1, x_2, x_3, Z) = (x_0, x_1, x_2, x_3, Z)$$

- The operator P_1 defines customer assignment to server 1 and is given by

$$P_1(x_0, x_1, x_2, x_3, O) = \begin{cases} (x_0 - 1, x_1 + 1, x_2, x_3, O), & \text{if } x_0 \geq 1, x_1 = 0 \\ & x_2 = 1, x_3 \geq 0 \\ & Z = O; \\ (x_0 - 1, x_1 + 1, x_2, x_3, I), & \text{if } x_0 \geq 1, x_1 = 0 \\ & x_2 = 0, x_3 \geq 0 \\ & Z = I, O \end{cases}$$

- The operator P_2 defines customer assignment to server 2 and is given by

$$P_2(x_0, x_1, x_2, x_3, I) = \begin{cases} (x_0 - 1, x_1, x_2 + 1, x_3, I), & \text{if } x_0 \geq 1, x_1 = 1 \\ & x_2 = 0, x_3 \geq 0 \\ & Z = I; \\ (x_0 - 1, x_1, x_2 + 1, x_3, O), & x_0 \geq 1, x_1 = 0; \\ & x_2 = 0, x_3 \geq 0 \\ & Z = I, O; \end{cases}$$

- Finally, the operator P_b defines customer assignment to both the servers at the same time and is given by,

$$P_b(x_0, x_1, x_2, x_3, Z) = (x_0 - 2, x_1 + 1, x_2 + 1, x_3, Z)$$

$$\text{if} \quad x_0 \geq 2, x_1 = 0, x_2 = 0, x_3 \geq 0, Z = I \quad or \quad O$$

Let $U = \{u = (u_0, u_1, u_2) : u_i \in \{h, 1, 2, b\}\}$ be the set of available controls where control action u_1 is to be taken on the occurrence of event A and the control action $u_i, i = 1, 2$ is to be taken on the occurrence of event D_i. Let

$$U(x) = \{u \in U : A(x) \in Dom(P_{u_0}), D_i(x) \in Dom(P_{u_i}), i = 1, 2\} \tag{5.6}$$

be the set of admissble controls when the system state is x. Note that $U(x)$ can be expressed as the cartesian product

$$U(x) = U_0(x) \times U_1(x) \times U_2(x) \tag{5.7}$$

where

$$\begin{aligned} U_0(x) =& \{u_0 : A(x) \in Dom(P_{u_0})\} \\ U_i(x) =& \{u_i : D_i(x) \in Dom(P_{u_i})\} \quad i = 1, 2 \end{aligned}$$

In order to complete the specification of the Markov decision process, define the transition probability function of the discrete time Markov chain as,

$$\begin{aligned} P(x(t+1) = y \mid x(t) = x, u(t) = u) &= \lambda, \qquad \text{if } y = P_{u_0} Ax \\ &= \mu_i, \qquad \text{if } y = P_{u_i} D_i x \end{aligned} \tag{5.8}$$

The model operates as follows, if the present state is x, then a control policy (u_0, u_1, u_2) is chosen such that if an arrival occurs, policy P_{u_0} is employed; if a departure from server i occurs then then policy P_{u_i} is employed.

5.6 The Dynamic Programming Formulation.

Consider the discounted cost criterion in discrete time

$$E[\sum_{t=0}^{\infty} \beta^t (x_0 + x_1 + x_2 + x_3)] \tag{5.9}$$

where $\frac{\beta = c}{(c+\alpha)}$ is a discount factor. This equation is an easy consequence of discretizing the continous time cost criterion of Section 5.3 by the uniformization procedure of Section 5.4.

Define a stationary policy π as a function $\pi : E \to U$ with $\pi(x) \in U(x)$ for every $x \in X$. When a stationary policy π is used, the control $u = \pi(x)$ is applied whenever the system is in state x.

Define by the space F the collection of all functions $f : E \to R$ so that the norm $||.||$ defined as

$$||f|| = \sup_{x \in X} \frac{f(x)}{\max(x_0 + x_1 + x_2 + x_3, 1)} \tag{5.10}$$

is finite. For any stationary policy π, define the dynamic programming operator by

$$T_\pi f(x) = x_0 + x_1 + x_2 + x_3 + \beta\lambda f(P_{u_0} Ax) + \beta\mu_1 f(P_{u_1} D_1 x) + \beta\mu_2 f(P_{u_2} D_2 x) \tag{5.11}$$

for all x in E, where $\pi(x) = (u_0, u_1, u_2)$. The dynamic programming operator T is now defined by

$$(Tf)(x) = \min_\pi (T_\pi f)(x) \tag{5.12}$$

Some well known results are stated next for future reference [47].

- For some n , $T^{(n)}$ is a contraction.
- If J^β is the optimal cost function then $J^\beta = TJ^\beta$.
- For any $f \in F, \lim_{n\to\infty} T^{(n)} f = J^\beta$
- There always exists an optimal policy π which is stationary.
- A stationary policy π is optimal iff $J^\beta = T_\pi J^\beta$

The dynamic programming equation can be written as

$$J^\beta(x) = \min_{u \in U(x)} [x_0 + x_1 + x_2 + x_3 + \beta\lambda J^\beta(P_{u_0} Ax) + \beta\mu_1 J^\beta(P_{u_1} D_1 x) + \beta\mu_2 J^\beta(P_{u_2} D_2 x)] \tag{5.13}$$

It is plain from (5.7) that the minimization over u in (5.13) can be performed by doing separate minimizations over u_0, u_1 and u_2. Therefore we can rewrite the DP equation as

$$J^\beta(x) = x_0 + x_1 + x_2 + x_3 + \min_{u_0 \in U_0(x)} \beta\lambda J^\beta(P_{u_0} Ax) + \min_{u_1 \in U_1(x)} \beta\mu_1 J^\beta(P_{u_1} D_1 x) + \min_{u_2 \in U_2(x)} \beta\mu_2 J^\beta(P_{u_2} D_2 x) \tag{5.14}$$

5.6 Identification of the Optimal Control of the Faster Server

In the next two sections, several equations that are satisfied by the optimal value function are stated and an interpretation about their significance is provided. Their proofs however, being quite tedious are given in the next chapter.

In the present section, we identify the inequalities that the value function must satisfy in order to specify the optimal control of the faster server.

Lemma 5.6.1. *The following inequalities*

$$f(P_h x) \geq f(P_1 x), \quad if \ x \in Dom(P_1) \tag{5.15}$$

and

$$f(P_2 x) \geq f(P_1 x), \quad if \ x \in Dom(P_1 \ and \ Dom(P_2) \tag{5.16}$$

are propagated under the dynamic programming operator.

∎

An interpretation of Lemma 5.6.1 is now provided. Inequality (5.15) implies that if server 1 is idle, then it is always optimal to assign a customer to it, if one is available in the buffer, irrespective of whether server 2 is busy or idle. Inequality (5.16) implies that if both servers are idle then it is optimal to assign a customer to server 1 rather than to server 2. Collectively, they tell us that it is always optimal to keep server 1 busy whenever possible. These conclusions are summarised in the following theorem.

Theorem 5.6.1. *Whenever server 1 is idle, it is optimal to assign to it a customer if one is waiting for service.*

∎

In order to show that the optimal value function satisfies (5.15)-(5.16), it is sufficient to prove that these equations are propagated under the dynamic programming operator, in other words Tf also satisfies (5.15)-(5.16). In the process of doing so, we found that it was necessary for the value function to satisfy additional properties, which are stated in Lemma 5.6.2.

Lemma 5.6.2. *The following inequalities (5.17)-(5.22) are propagated under the dynamic programming operator, where inequalities*

$$f(x_0,0,1,x_3,O) \geq f(x_0-1,1,1,x_3,O) \quad x_0 \geq 1, x_3 \geq 0 \tag{5.17a}$$

$$f(x_0,0,0,0,Z) \geq f(x_0,1,0,0,I) \quad x_0 \geq 0 \tag{5.17b}$$

$$f(x_0,0,1,0,O) \geq f(x_0,1,0,0,I) \quad x_0 \geq 0 \tag{5.17c}$$

$$f(x_0+1,x_1,x_2,x_3,Z) \geq f(x_0,x_1,x_2,x_3,Z) \quad x_0,x_3 \geq 0, x_1,x_2 \in \{0,1\} \tag{5.18a}$$

$$f(x_0,x_1,x_2,x_3+1,Z) \geq f(x_0,x_1,x_2,x_3,Z) \quad x_0,x_3 \geq 0, x_1,x_2 \in \{0,1\} \tag{5.18b}$$

$$f(x_0,1,1,x_3,I) \geq f(x_0,1,0,x_3,I) \quad x_0,x_3 \geq 0 \tag{5.18c}$$

$$f(x_0,1,1,x_3,O) \geq f(x_0,0,1,x_3,O) \quad x_0,x_3 \geq 0, \tag{5.18d}$$

$$f(x_0,1,0,x_3,I) \geq f(x_0,0,0,0,I) \quad x_0,x_3 \geq 0 \tag{5.18e}$$

$$f(x_0,0,1,x_3,O) \geq f(x_0,0,0,O) \quad x_0,x_3 \geq 0 \tag{5.18f}$$

$$f(x_0,1,1,x_3,O) \geq f(x_0,0,1,x_3+1,O) \quad x_0,x_3 \geq 0 \tag{5.19a}$$

$$f(x_0,1,1,x_3,O) \geq f(x_0-1,1,1,x_3+1,O) \quad x_0 \geq 1, x_3 \geq 0 \tag{5.19b}$$

$$f(x_0,1,1,0,O) \geq f(x_0,1,1,0,I) \quad x_0 \geq 0 \tag{5.20}$$

$$f(x_0,1,1,x_3,O) - f(x_0,1,1,x_3',O) \geq f(y_0,1,1,y_3,I) - f(y_0,1,1,y_3',I)$$
$$x_0,x_0',x_3,x_3',y_3,y_3' \geq 0, x_3-x_3' \geq y_3-y_3' \tag{5.21a}$$

$$f(x_0,1,1,x_3,O) - f(x_0,1,1,x_3',O) \geq f(y_0,1,0,y_3,I) - f(y_0,1,0,y_3',I)$$
$$x_0,y_0,x_3,x_3',y_3,y_3' \geq 0, x_3-x_3' \geq y_3-y_3' \tag{5.21b}$$

$$f(x_0,0,1,x_3,O) - f(x_0,0,1,x_3',O) \geq f(y_0,1,0,y_3,I) - f(y_0,1,0,y_3',O)$$
$$x_0,y_0,x_3,x_3',y_3,y_3' \geq 0, x_3-x_3' \geq y_3-y_3' \tag{5.21c}$$

$$f(x_0,0,1,x_3,O) - f(x_0,01,x_3',O) \geq f(y_0,1,1,y_3,I) - f(y_0,1,1,y_3',O)$$
$$x_0,y_0,x_3,x_3',y_3,y_3' \geq 0, x_3-x_3' \geq y_3-y_3' \tag{5.21d}$$

$$f(x_0,1,0,x_3,I)-f(x_0-1,1,1,x_3,I) = f(x_0,1,0,x_3+1,I)-f(x_0-1,1,1,x_3+1,I)$$
$$x_0 \geq 1, x_3 \geq 0 \tag{5.22a}$$

$$f(x_0,1,1,x_3,I) - f(x_0-1,1,1,x_3+1,I)$$
$$= f(x_0,1,1,x_3+1,I) - f(x_0-1,1,1,x_3+2,I) \qquad x_0 \geq 1, x_3 \geq 0 \quad (5.22b)$$
$$f(x_0,1,0,x_3,I) - f(x_0-1,1,0,x_3+1,I)$$
$$= f(x_0,1,0,x_3+1,I) - f(x_0-1,1,0,x_3+2,I) \qquad x_0 \geq 1, x_3 \geq 0 \quad (5.22c)$$
$$f(x_0,1,1,x_3,I) - f(x_0,1,0,x_3+1,I)$$
$$= f(x_0,1,1,x_3+1,I) - f(x_0,1,0,x_3+2,I) \qquad x_0 \geq 0, x_3 \geq 0 \quad (5.22d)$$
$$f(x_0,1,0,x_3+1,I) - f(x_0-2,1,1,x_3+2,I)$$
$$= f(x_0,1,0,x_3,I) - f(x_0-2,1,1,x_3+1,I) \qquad x_0 \geq 2, x_3 \geq 0 \quad (5.22e)$$

∎

It is shown in the next chapter that if f satisfies (5.17)-(5.22), then Tf also satisfies them, so that these propeties are invariant under the dynamic programming operator.

An intuitive interpretation is now provided for some these propeties. First note that (5.17a-c) are the same as (5.15)-(5.16) of Lemma 5.6.1. Inequalities (5.18a-f) express the monotonicity properties of the optimal value function. The value function increases if the magnitude of any one the first four states is increased, the other states remaining the same. This should be intuitively clear since the cost is linear in the first four states.

Inequalities (5.19a-b) are also interesting. Equation (5.19b) reveals the fact if the total number of customers in the main queue buffer and the resequencing buffer is kept constant and as the number in the queue buffer is decreased and that in the resequencing buffer is increased, the value function decreases. Hence it is 'better' to have a customer in the resequencing buffer rather than in the queue buffer. This is also intuitively appealing since a customer in the resequencing buffer has already finished its service, while the one in the queue buffer is yet to receive service. Equation (5.19a) is subject to a similar interpretation.

Equation (5.20) states that the value function is larger for an out-of -sequence state compared to an in-sequence state. An intuitive explanation for this is that when $Z = O$, then there there is a greater potential for a large number of customers to accumulate in the resequencing buffer, compared to the case when $Z = I$. Equations (5.21a-d) were found to be necassary to make (5.20) propagate.

Equation (5.22a) indicates that the decision to allocate a customer to the slower server is independent of the number of customers in the resequencing buffer. This can be shown as follows. Suppose that there are k customers in the queue buffer and n customers in the resequencing buffer, and that it is optimal to assign a customer to the slower server, in which case

$$f(x_0, 1, 0, x_3, I) - f(x_0 - 1, 1, 1, x_3, I) \geq 0$$

But by (5.22a) this implies that

$$f(x_0, 1, 0, x_3 + 1, I) - f(x_0 - 1, 1, 1, x_3 + 1, I) \geq 0$$

Hence it also optimal to assign a customer to the slower server when there are $n + 1$ customers in the resequencing buffer. Equations (5.22c-e) were found to be necessary to make (5.22a) propagate.

5.7 Identification of the Optimal Control of the Slower Server

From (5.22a) in the last section, we already know that the decision to assign customers to the slower server is independent of the number of customers in the resequencing buffer. In this section we complete the specification of the optimal control of the slower server by showing that it is of the threshold type in the number of customers in the main queue buffer when there are no customers in the resequencing buffer. A proof by policy iteration was attempted but did not succeed. The proof given by Maglaris [50] was found to be applicable with a few variations.

Before stating the main result, we need the following lemma.

Lemma 5.7.1 *The following inequalities are propagated under, where the dynamic programming operator,*

$$f(x_0,1,1,0,O) - f(x_0-1,1,1,O) = f(x_0,1,1,0,I) - f(x_0-1,1,1,0,I)$$

$$x_0 \geq 0 \quad (5.23a)$$

$$f(x_0,1,1,x_3,Z) - f(x_0-1,1,1,x_3,Z) = f(x_0,1,1,x_3-1,I) - f(x_0-1,1,1,x_3-1,I)$$

$$x_0, x_3 \geq 1 \quad (5.23b)$$

$$f(x_0,1,0,x_3,I) - f(x_0-1,1,0,x_3',I) = f(y_0,1,0,y_3-1,I) - f(y_0-1,1,0,y_3'-1,I)$$

$$x_0, y_0, x_3, x_3', y_3, y_3' \geq 0, x_3 - x_3' = y_3 - y_3' \quad (5.23c)$$

$$f(x_0,1,1,x_3,I) - f(x_0,1,1,x_3',I) - f(y_0,1,0,y_3,I) - f(y_0,1,0,y_3',I)$$

$$x_0, y_0, x_3, x_3', y_3, y_3' \geq 0, x_3 - x_3' = y_3 - y_3' \quad (5.23d)$$

$$f(x_0,1,1,x_3,O) - f(x_0,1,1,x_3',O) = f(y_0,0,1,y_3,O) - f(y_0,0,1,y_3',O)$$

$$x_0, y_0, x_3, x_3', y_3, y_3' \geq 0, x_3 - x_3' = y_3 - y_3' \quad (5.23e)$$

∎

Equation (5.23a) is crucial for the success of the next result, which is adapted from a proof in [50]. Equations (5.23b-e) were found to be necessary to (5.23a) propagate.

Before stating the next result, we need a few more definitions.

Let f_n^β be the function obtained from the nth iteration of the dynamic programming equation. For all $n = 0, 1\ldots$ define $\{\Delta_n(i)\}_2^\infty$ and $\{\tilde{\Delta}_n(i)\}_2^\infty$ as by

$$\tilde{\Delta}_n(1) = f_n(0,1,0,0,I) - f_n(0,0,0,0,Z)$$

$$\tilde{\Delta}_n(i) = f_n(i-1,1,0,0,I) - f_n(i-2,1,0,0,I) \qquad i \geq 2 \qquad n = 0, 1\ldots$$

$$\Delta_n(2) = f_n(0,1,1,0,I) - f_n(0,0,1,0,O)$$

$$\Delta_n(i) = f_n(i-2,1,1,0,I) - f_n(i-3,1,1,0,I) \qquad i > 2 \qquad n = 0, 1\ldots$$

We now state the main result of this section.

Lemma 5.7.1. *The inequalities*

$$\Delta_n(i) \leq \tilde{\Delta}_n(i) \qquad i \geq 2 \qquad n = 0, 1\ldots (5.24a)$$

$$\Delta_n(i) \geq \Delta_n(i-1) \geq 0 \qquad i \geq 2 \qquad n = 0, 1\ldots (5.24b)$$

$$\tilde{\Delta}_n(i) \geq \tilde{\Delta}_n(i-1) \geq 0 \qquad i \geq 2 \qquad n = 0, 1\ldots (5.24c)$$

$$\Delta_n(i) \geq \tilde{\Delta}_n(i-1) \qquad i \geq 2 \qquad n = 0, 1\ldots (5.24d)$$

hold true for all $i \geq 2$. ∎

In order to make (5.24a-d) propagate, it is essential that (5.23a) be satisfied. We now discuss the significance of these relations. Inequality (5.24a) implies a threshold policy as shown below. By (5.24a), it is plain that for all $i \geq 2$,

$$f_n(i-2,1,1,0,I) - f_n(i-1,1,0,0,I) \leq f_n(i-3,1,1,0,I) - f_n(i-2,1,0,0,I)$$

whence at the n^{th} iteration,

$$f_n(0,1,1,0,I) - f_n(1,1,0,0,I) \geq$$

$$f_n(1,1,1,0,I) - f_n(2,1,0,0,I) \geq \dots$$

$$\dots \geq f_n(K-2,1,1,0,I) - f_n(K-1,1,0,0,I) > 0 > \dots$$

$$\dots > 0 \geq f_n(K-1,1,1,0,I) - f_n(K,1,0,0,I) \geq$$

$$f_n(K,1,1,0,I) - f_n(K+1,1,0,0,I) \geq \dots$$

Hence since $f_n(i-1,1,1,0,I) - f_n(i,1,0,0,I)$ increases monotonically in n, it crosses zero for some value of i, say K. This implies that the threshold is K, for zero customers in the resequencing buffer.

Equations (5.24b) implies that $\Delta_n(i)$ is non decreasing and convex in i, while (5.24c) implies that $\tilde{\Delta}_n(i)$ is non decreasing and convex in i.

These conclusions are summarized in the next theorem.

Theorem 5.7.1 *The optimal rule for assigning customers to the slower server is independent of the number of customers in the resequencing buffer, and is of the threshold type in the number of customers in the main queue buffer*

∎

In the next chapter the proofs of all the lemmas in the last two sections are provided.

CHAPTER VI

PROOFS OF THE MAIN RESULTS

6.1 Introduction

In the present chapter we provide the proofs for Lemmas 5.6.2, 5.7.1 and 5.7.2 from Chapter 5. All the proofs proceed using value iteration and amount to showing that the property under investigation is propagated by the dynamic programming operator.

The rest of the chapter is organized as follows. In Section 6.1 we provide a proof for Lemma 5.6.2, while in Section 6.2 we provide proofs for Lemmas 5.7.1 and 5.7.2.

6.2 Optimal Control of the Faster Server

In this section, we provide a proof for Lemma 5.6.2, which as the reader will recall, leads to the conclusion that the faster server should be kept busy whenever possible. The proof proceeds in three steps. We first provide a proof for the group (5.22a-e) in Lemma 6.2.2 which is independent of the proofs of the other equations. We then provide a proof for the group (5.21a-d) in Lemma 6.2.3 and utilise Lemma 6.2.2 in the course of the proof. Finally using Lemmas 6.2.1-2, we provide a proof for (5.17a-b), (5.18a-f), (5.19a-b) and (5.20) in Lemma 6.2.3 which concludes the discussion.

Lemma 6.2.1. *Equations (5.22a-e) of Lemma 5.6.2 propagate under the dynamic programming operator, i.e.,*

$$Tf(x_0,1,0,x_3,I) - Tf(x_0-1,1,1,x_3,I)$$

$$= Tf(x_0,1,0,x_3+1,I) - Tf(x_0-1,1,1,x_3+1,I) \qquad x_0 \geq 1, x_3 \geq 0 \quad (6.1a)$$

$$Tf(x_0,1,1,x_3,I) - Tf(x_0-1,1,1,x_3+1,I)$$

$$= Tf(x_0,1,1,x_3+1,I) - Tf(x_0-1,1,x_3+2,I) \qquad x_0 \geq 1, x_3 \geq 0 \quad (6.1b)$$

$$Tf(x_0,1,0,x_3,I) - Tf(x_0-1,1,0,x_3+1,I)$$

$$= Tf(x_0,1,0,x_3+1,I) - Tf(x_0-1,1,0,x_3+2,I) \qquad x_0 \geq 1, x_3 \geq 0 \quad (6.1c)$$

$$Tf(x_0,1,1,x_3,I) - Tf(x_0,1,0,x_3+1,I)$$

$$= Tf(x_0,1,1,x_3+1,I) - Tf(x_0,1,0,x_3+2,I) \qquad x_0 \geq 0, x_3 \geq 0 \quad (6.1d)$$

$$Tf(x_0,1,0,x_3+1,I) - Tf(x_0-2,1,1,x_3+2,I)$$

$$= Tf(x_0,1,0,x_3,I) - Tf(x_0-2,1,1,x_3+1,I) \qquad x_0 \geq 2, x_3 \geq 0 \quad (6.1e)$$

Proof. We provide proofs for (6.5a-c) with the proofs for (6.5d-e) left to the reader.

- **To Prove That**

$$Tf(x_0,1,0,x_3,I) - Tf(x_0-1,1,1,x_3,I)$$

$$= Tf(x_0,1,0,x_3+1,I) - Tf(x_0-1,1,1,x_3+1,I) \qquad (6.1a)$$

In view of (5.14),equation (6.1a) holds if equations (6.2),(6.3) and (6.4) given below are satisfied.

-

$$\min_{h,2} f(P_{u_0}(x_0+1,1,0,x_3+1,I)) - f(P_{u_0}(x_0,1,1,x_3+1,I))$$

$$= \min_{h,2} f(P_{u_0}(x_0+1,1,0,x_3,I)) - f(P_{u_0}(x_0,1,1,x_3,I)) \qquad (6.2)$$

If we apply the hold operator on both sides of (6.2), then it is true because of (5.22a). If we apply the operator P_2, then both sides of the equation reduce to zero.

-

$$\min_{1,b} f(P_{u_1}(x_0,0,0,0,O)) - f(x_0-2,0,1,0,O)$$

$$= \min_{1,b} f(P_{u_1}(x_0,0,0,0,O)) - f(x_0-2,0,1,0,O)) \qquad (6.3)$$

Equation (6.3) is obviouisly true by inspection.

-

$$\min_{h,2} f(P_{u_2}(x_0,1,0,x_3,I)) - \min_{h,2} f(P_{u_2}(x_0-1,1,0,x_3+1,I))$$

$$= \min_{h,2} f(P_{u_2}(x_0,1,0,x_3+1,I)) - \min_{h,2} f(P_{u_2}(x_0-1,1,0,x_3+2,I)) \quad (6.4)$$

Expanding (6.4), we obtain

$$\min[f(x_0,1,0,x_3+1,I), f(x_0-1,1,1,x_3+1,I)]$$

$$-\min[f(x_0-1,1,0,x_3+2,I), f(x_0-2,1,1,x_3+2,I)]$$

$$= \min[f(x_0,1,0,x_3,I), f(x_0-1,1,1,x_3,I)]$$

$$-\min[f(x_0-1,1,0,x_3+1,I), f(x_0-2,1,1,x_3+1,I)] \qquad (6.4')$$

and (6.4′) is [illegible] true if the following equations hold:

$$f(x_0, 1, 0, x_3 + 1, I) - f(x_0 - 1, 1, 0, x_3 + 2, I)$$

$$= f(x_0, 1, 0, x_3, I) - f(x_0 - 1, 1, 0, x_3 + 1, I) \tag{6.5}$$

$$f(x_0 - 1, 1, 1, x_3 + 1, I) - f(x_0 - 2, 1, 1, x_3 + 2, I)$$

$$= f(x_0 - 1, 1, 1, x_3, I) - f(x_0 - 2, 1, 1, x_3 + 1, I) \tag{6.6}$$

$$f(x_0 - 1, 1, 1, x_3 + 1, I) - f(x_0 - 1, 1, 0, x_3 + 2, I)$$

$$= f(x_0 - 1, 1, 1, x_3, I) - f(x_0 - 1, 1, 0, x_3 + 1, I) \tag{6.7}$$

and

$$f(x_0, 1, 0, x_3 + 1, I) - f(x_0 - 2, 1, 1, x_3 + 2, I)$$

$$= f(x_0, 1, 0, x_3, I) - f(x_0 - 2, 1, 1, x_3 + 1, I) \tag{6.8}$$

But (6.5)-(6.8) are just equations $(5.22c - e)$ which are assumed to be true. Hence equation (6.1a) is proved.

- **To Prove That**

$$Tf(x_0, 1, 1, x_3, I) - Tf(x_0 - 1, 1, 1, x_3 + 1, I)$$

$$= Tf(x_0, 1, 1, x_3 + 1, I) - Tf(x_0 - 1, 1, 1, x_3 + 2, I) \tag{6.1b}$$

This equation is satisfied provided (6.9), (6.10) and (6.11) hold.

∘

$$f(x_0 + 1, 1, 1, x_3, I) - f(x_0, 1, 1, x_3 + 1, I)$$

$$= f(x_0 + 1, 1, 1, x_3 + 1, I) - f(x_0, 1, 1, x_3 + 2, I) \tag{6.9}$$

This is true because of $(5.22b)$.

∘

$$\min_{h,1} f(P_{u_1}(x_0, 0, 1, 0, O)) - \min_{h,1} f(P_{u_1}(x_0 - 1, 0, 1, 0, O))$$

$$= \min_{h,1} f(P_{u_1}(x_0, 0, 1, 0, O)) - \min_{h,1} f(P_{u_1}(x_0 - 1, 0, 1, 0, O)) \tag{6.10}$$

Equation (6.10) is obviously true by inspection.

○

$$\min_{h,2} f(P_{u_2}(x_0, 1, 0, x_3 + 1, I)) - \min_{h,2} f(P_{u_2}(x_0 - 1, 1, 0, x_3 + 2, I))$$

$$= \min_{h,2} f(P_{u_2}(x_0, 1, 0, x_3 + 2, I)) - \min_{h,2} f(P_{u_2}(x_0 - 1, 1, 0, x_3 + 3, I)) \quad (6.11)$$

Equation (6.11) is true because of (5.22b-e).

- **To Prove That**

$$Tf(x_0, 1, 0, x_3, I) - Tf(x_0 - 1, 1, 0, x_3 + 1, I)$$

$$= Tf(x_0, 1, 0, x_3 + 1, I) - Tf(x_0 - 1, 1, 0, x_3 + 2, Ia \qquad (6.1c)$$

This equation will hold provided the following equations (6.12), (6.13) and (6.14) are satisfied.

○

$$\min_{h,2} f(P_{u_0}(x_0 + 1, 1, 0, x_3, I)) - \min_{h,2} f(P_{u_0}(x_0, 1, 0, x_3 + 1, I))$$

$$= \min_{h,2} f(P_{u_0}(x_0 + 1, 1, 0, x_3 + 1, I)) - \min_{h,2} f(P_{u_0}(x_0, 1, 0, x_3 + 2, I)) \quad (6.12)$$

Equation (6.12) is easily seen to hold because of (5.22c).

○

$$\min_{1,b} f(P_{u_1}(x_0, 0, 0, 0, O)) - \min_{1,b} f(P_{u_1}(x_0 - 1, 0, 0, 0, I))$$

$$= \min_{1,b} f(P_{u_1}(x_0, 0, 0, 0, I)) - \min_{1,b} f(P_{u_1}(x_0 - 1, 0, 0, 0, I)) \qquad (6.13)$$

Equation (6.13) is obviously true by inspection

○

$$\min_{2,h} f(P_{u_2}(x_0, 1, 0, x_3, I)) - \min_{h,2} f(P_{u_2}(x_0 - 1, 1, 0, x_3 + 1, I))$$

$$= \min_{h,2} f(P_{u_2}(x_0, 1, 0, x_3 + 1, I)) - \min_{h,2} f(P_{u_2}(x_0 - 1, 1, 0, x_3 + 2, I)) \quad (6.14)$$

Equation (6.14) is also easily seen to be true from equations $(5.22b - e)$.

Equations ($6.1d - e$) have a very similar derivation and we leave them to the reader.

∎

In the next lemma we provide proofs for equations (5.21a-d).

Lemma 6.2.2. *Equations (5.21a-d) propagate under the dynamic programming operator, i.e.,*

$$Tf(x_0,1,1,x_3,O) - Tf(x_0,1,1,x_3',O) \geq Tf(y_0,1,1,y_3,I) - Tf(y_0,1,1,y_3',I)$$

$$x_0, y_0, x_3, x_3', y_3, y_3' \geq 0, x_3 - x_3' \geq y_3 - y_3' \quad (6.15a)$$

$$Tf(x_0,1,1,x_3,O) - Tf(x_0,1,1,x_3',O) \geq Tf(y_0,1,0,y_3,I) - Tf(y_0,1,0,y_3',I)$$

$$x_0, y_0, x_3, x_3', y_3, y_3' \geq 0, x_3 - x_3' \geq y_3 - y_3' \quad (6.15b)$$

$$Tf(x_0,0,1,x_3,O) - Tf(x_0,0,1,x_3',O) \geq Tf(y_0,1,0,y_3,I) - Tf(y_0,1,0,y_3',O)$$

$$x_0, y_0, x_3, x_3', y_3, y_3' \geq 0, x_3 - x_3' \geq y_3 - y_3' \quad (6.15c)$$

$$Tf(x_0,0,1,x_3,O) - Tf(x_0,0,1,x_3',O) \geq Tf(y_0,1,1,y_3,I) - Tf(y_0,1,1,y_3',O)$$

$$x_0, y_0, x_3, x_3', y_3, y_3' \geq 0, x_3 - x_3' \geq y_3 - y_3' \quad (6.15d)$$

Proof. We provide a proof for (6.15a), and since the proofs for (6.15c-d) are similar, we leave them to the interested reader.

- **To Prove That**

$$Tf(x_0,1,1,x_3,O) - Tf(x_0,1,1,x_3',O)$$

$$\geq Tf(y_0,1,1,y_3,I) - Tf(y_0,1,1,y_3',I)$$

This equation will be true provided (6.16)-(6.17) below are satisfied.

∘

$$f(x_0+1,1,1,x_3,O) - f(x_0+1,1,1,x_3',O)$$

$$\geq f(y_0+1,1,1,y_3,I) - f(y_0+1,1,1,y_3',I) \quad (6.16)$$

Equation (6.16) follows by (5.21a).

∘

$$\min_{h,1} f(x_0,0,1,x_3+1,O) - \min_{h,1} f(x_0,0,1,x_3'+1,O)$$

$$\geq \min_{h,2} f(y_0,1,0,y_3+1,I) - \min_{h,2} f(y_0,1,0,y_3'+1,O) \tag{6.17}$$

Equation (6.17) holds because of (5.21a-d) and this concludes the proof of (6.15a).

∎

In the next lemma we provide the proofs for the rest of the equations in Lemma 5.6.2. Since the proofs of each one of them depends upon the others, we present all their proofs in the same lemma.

Lemma 6.2.3. *The inequalities (5.17a-c), (5.18a-f), (5.19a-b) and (5.20) propagate under the dynamic programming operator, i.e.,*

$$Tf(x_0,0,1,x_3,O) \geq Tf(x_0-1,1,1,x_3,O) \quad x_0 \geq 1, x_3 \geq 0 \tag{6.18a}$$

$$Tf(x_0,0,0,0,Z) \geq Tf(x_0,1,0,0,I) \quad x_0 \geq 0 \tag{6.18b}$$

$$Tf(x_0,0,1,0,O) \geq Tf(x_0,1,0,0,I) \quad x_0 \geq 0 \tag{6.18c}$$

$$Tf(x_0+1,x_1,x_2,x_3,Z) \geq Tf(x_0,x_1,x_2,x_3,Z)$$

$$x_0,x_3 \geq 0, x_1,x_2 \in \{0,1\} \tag{6.19a}$$

$$Tf(x_0,x_1,x_2,x_3+1,Z) \geq Tf(x_0,x_1,x_2,x_3,Z)$$

$$x_0,x_3 \geq 0, x_1,x_2 \in \{0,1\} \tag{6.19b}$$

$$Tf(x_0,1,1,x_3,I) \geq Tf(x_0,1,0,x_3,I) \quad x_0,x_3 \geq 0 \tag{6.19c}$$

$$Tf(x_0,1,1,x_3,O) \geq Tf(x_0,0,1,x_3,O) \quad x_0,x_3 \geq 0, \tag{6.19d}$$

$$Tf(x_0,1,0,x_3,I) \geq Tf(x_0,0,0,0,I) \quad x_0,x_3 \geq 0 \tag{6.19e}$$

$$Tf(x_0,0,1,x_3,O) \geq Tf(x_0,0,0,0,O) \quad x_0,x_3 \geq 0 \tag{6.19f}$$

$$Tf(x_0,1,1,x_3,O) \geq Tf(x_0,0,1,x_3+1,O) \quad x_0,x_3 \geq 0 \tag{6.20a}$$

$$Tf(x_0,1,1,x_3,O) \geq Tf(x_0-1,1,1,x_3+1,O) \quad x_0 \geq 1, x_3 \geq 0 \tag{6.20b}$$

$$Tf(x_0,1,1,0,O) \geq Tf(x_0,1,1,0,I) \tag{6.21}$$

Proof. We provide proofs for (6.18a-c), (6.19a) and (6.19c), (6.20a) and (6.21). The other proofs are left to the reader.

- **To Prove That**

$$Tf(x_0,0,1,x_3,O) \geq Tf(x_0-1,1,1,x_3,O) \qquad x_0 \geq 1 \tag{6.18a}$$

We will consider the case $x_0 \geq 2$. The following inequalities (6.22), (6.23) and (6.24) have to be verified.

∘

$$\min_{h,1} f(P_{u_0}(x_0+1,0,1,x_3,O)) \geq \min_{h} f(P_{u_0}(x_0,1,1,x_3,O)) \tag{6.22}$$

By (5.17a) this reduces to the comparison

$$f(x_0,1,1,x_3,O) \geq f(x_0,1,1,x_3,O)$$

which holds with equality.

∘

$$\min_{h,1} f(P_{u_1}(x_0,0,1,x_3,O) \geq \min_{h,1} f(P_{u_1}(x_0-1,0,1,x_3+1,O) \tag{6.23}$$

By (5.17a) this reduces to

$$f(x_0-1,1,1,x_3,O) \geq f(x_0-2,1,1,x_3+1,O)$$

which is true by (5.19b).

∘

$$\min_{1,b,h} f(P_{u_2}(x_0,0,0,0,I)) \geq \min_{2,h} f(P_{u_2}(x_0-1,1,0,0,I)) \tag{6.24}$$

If we take action P_1 on the left-hand-side and P_h on the right-hand-side, (6.24) reduces to an equality. The other cases can be treated in a similar fashion.

- **To Prove That**

$$Tf(x_0,0,0,0,I) \geq Tf(x_0-1,1,0,0,I) \qquad x_0 \geq 1 \tag{6.18b}$$

The following inequalities (6.25), (6.26) and (6.27) have to be verified.

∘

$$\min_{1,b,h} f(P_{u_0}(x_0+1,0,0,0,I)) \geq \min_{2,h} f(P_{u_0}(x_0,1,0,0,I)) \tag{6.25}$$

Equation (6.25) is easily seen to be true because of (5.17*b*).

∘

$$\min_{1,b} f(P_{u_1}(x_0,0,0,0,I)) \geq \min_{1,b} f(P_{u_1}(x_0-1,0,0,0,I)) \tag{6.26}$$

Equation (6.26) holds by (5.18*a*).

∘

$$\min_{1,b,h} f(P_{u_2}(x_0,0,0,0,I)) \geq \min_{h,2,h} f(P_{u_2}(x_0-1,1,0,0,I)) \tag{6.27}$$

Equation (6.27) holds by (5.17*b*).

- **To Prove That**

$$Tf(x_0-1,0,1,0,O) \geq Tf(x_0-1,1,0,0,I) \tag{6.18c}$$

We start by verifying equations (6.28-31).

∘

$$\min_{h,1} f(P_{u_0}(x_0,0,1,0,O)) \geq \min_{h,2} f(P_{u_0}(x_0,1,0,0,I)) \tag{6.28}$$

This reduces to

$$f(x_0-1,1,1,0,O) \geq \min(f(x_0-1,1,1,0,I), f(x_0,1,0,0,I))$$

If

$$f(x_0-1,1,1,0,I) \geq f(x_0,1,0,0,I)$$

then

$$f(x_0-1,1,1,0,O) \geq f(x_0-1,1,1,0,I)$$

by (5.20) from which it follows that

$$f(x_0-1,1,1,0,O) \geq f(x_0,1,0,0,I)$$

∘

$$\min_{b,1} f(P_{u_1}(x_0-1,0,0,0,I)) = \min_{b,1} f(P_{u_2}(x_0-1,0,0,0,O)) \tag{6.29}$$

Equation (6.29) holds with equality.

∘

$$\min_{2,h} f(P_{u_2}(x_0-1,1,0,0,I)) \le \min_{1,h} f(P_{u_1}(x_0-1,0,1,0,O)) \tag{6.30}$$

Equation (6.30) can be verified in the same way as equation (6.28).

∘

$$\min_{1,b} f(P_{u_2}(x_0-1,0,0,0,O)) \le \min_{1,h} f(P_{u_1}(x_0-1,0,1,0,O)) \tag{6.31}$$

This can be easily shown to be true.

If $x = (x_0,0,0,0,Z)$ then we can write equations (6.28) to (6.31) as

$$\min(f(P_{u_0}AP_1x)) \le \min(f(P_{u_0}AP_2x)) \tag{6.28'}$$

$$\min(f(P_{u_1}D_1P_1x)) = \min(f(P_{u_2}D_2P_2x)) \tag{6.29'}$$

$$\min(f(P_{u_2}D_2P_1x)) \le \min(f(P_{u_1}D_1P_2x)) \tag{6.30'}$$

$$\min(f(P_{u_2}D_2P_2x)) \le \min(f(P_{u_1}D_1P_2x)) \tag{6.31'}$$

Hence

$$Tf(P_1x) = x_0 + \beta\lambda \min f(P_{u_0}AP_1x) + \beta\mu_1 \min f(P_{u_1}D_1P_1x)$$

$$+\beta\mu_2 \min f(P_{u_2}D_2P_1x)$$

$$\le x_0 + \beta\lambda \min f(P_{u_0}AP_2x) + \beta\mu_1 \min f(P_{u_2}D_2P_2x)$$

$$+\beta\mu_2 \min f(P_{u_1}D_1P_2x)$$

$$\le x_0 + \beta\lambda \min f(P_{u_0}AP_2x) + \beta\mu_1 \min f(P_{u_1}D_1P_2x)$$

$$+\beta\mu_2 \min f(P_{u_2}D_2P_2x)$$

$$= Tf(P_2^0 x)$$

This proves equation (6.22c). Note that the fact that $\mu_1 > \mu_2$ was explicitly used in this argument.

- **To Prove That**

$$Tf(x_0+1, x_1, x_2, x_3, I) \geq Tf(x_0, x_1, x_2, x_3, I) \qquad x_0 \geq 0, x_3 \geq 0 \quad (6.19a)$$

We will consider the case $x_0 \geq 1$. Equation (6.19a) is true provided (6.32)-(6.34) given below are satisfied.

∘

$$\min_{u_0} f(P_{u_0}(x_0+2, x_1, x_2, x_3, I)) \geq \min_{u_0} f(P_{u_0}(x_0+1, x_1, x_2, x_3, I)) \quad (6.32)$$

It can easily be checked that for all possible combinations of x_1, x_2 and the control action P_{u_0} equation (6.32) remains true because of (5.18a).

∘

$$\min_{h,1} f(P_{u_1}(x_0+1, 0, 1, 0, O)) \geq \min_{h,1} f(P_{u_1}(x_0, 0, 1, 0, O)) \quad (6.33)$$

by (5.17a), the smaller one among the operators P_h and P_1 is P_1. Hence (6.33) reduces to

$$f(x_0, 1, 1, 0, O) \geq f(x_0-1, 1, 1, 0, O) \quad (6.33')$$

But the inequality (6.33′) is true from (5.18a).

∘

$$\min_{h,2} f(P_{u_2}(x_0+1, 1, 0, x_3+1, I)) \geq \min_{h,2} f(P_{u_2}(x_0, 1, 0, x_3+1, I)) \quad (6.34)$$

Suppose we decide to assign customers to server 2 on both sides of the inequality. Then the truth of the inequality follows from (5.18a). If we decide not to assign customers to either one of two servers then the truth of the inequality again follows from (5.18a). If we decide to assign a customer to server 2 on the left hand side but not on the right hand side then (6.34) follows from (5.18b).

- **To Prove That**

$$Tf(x_0,1,1,x_3,I) \geq Tf(x_0,1,0,x_3,I) \qquad x_0 \geq 0 \tag{6.19c}$$

We consider the case $x_0 \geq 1$. Inequality (6.19c) reduces to the following inequalities in (6.35), (6.36) and (6.37).

∘

$$\min_h f(P_{u_0}(x_0+1,1,1,x_3,I)) \geq \min_{h,2} f(P_{u_0}(x_0+1,1,0,x_3,I)) \tag{6.35}$$

If we apply the P_h operator to the left hand side, then (6.35) is true by (5.18b). If we apply the P_2 operator then (6.35) is true by (5.18a).

∘

$$\min_{h,1} f(P_{u_1}(x_0,0,1,0,O)) \geq \min_{h,1,2,b} f(P_{u_1}(x_0,0,0,0,O)) \tag{6.36}$$

On the right hand side, the minimum of the operators P_h and P_1, is P_1 by (5.17a). On the left hand side the minimum operator is one of the two P_1 or P_2 by (5.17a), (5.17b) and (5.17c). Using these facts (6.36) can be rewritten as

$$f(x_0-1,1,1,0,O) \geq \min[f(x_0-1,1,0,0,I), f(x_0-2,1,1,0,O)] \tag{6.36$'$}$$

By (5.20) and (5.18c),

$$f(x_0-1,1,1,0,O) = f(x_0-1,1,1,0,I) \geq f(x_0-1,1,0,0,I)$$

whereas from (5.18a),

$$f(x_0-1,1,1,0,O) \geq f(x_0-2,1,1,0,O)$$

, it follows that (6.36$'$) is satisfied .

∘

$$\min_{h,2} f(P_{u_2}(x_0,1,0,x_3+1,I)) \geq \min_{h,2} f(P_{u_2}(x_0,1,0,x_3,I)) \tag{6.37}$$

Inequality (6.37) is true by (5.18b).

This concludes the proof of (6.19c).

Inequalities (6.23c), (6.23d) and (6.23e) can be verified in a similar fashion and their proof is omitted. We now go on to (6.24a).

- **To Prove That**

$$Tf(x_0, 1, 1, x_3, O) \geq Tf(x_0, 0, 1, x_3 + 1, O) \tag{6.20}$$

We will consider the case $x_0 \geq 2$. Inequality (6.20) will hold provided the following inequalities in (6.38), (6.39) and (6.40) are satisfied.

∘

$$f(P_{u_0}(x_0 + 1, 1, 1, x_3, O)) \geq \min_{h,1} f(P_{u_0}(x_0 + 1, 0, 1, x_3 + 1, O)) \tag{6.38}$$

By (5.3a), (6.38) reduces to

$$f(x_0 + 1, 1, 1, x_3, O) \geq f(x_0, 1, 1, x_3 + 1, O)$$

which is true because of (5.19b).

∘

$$\min_{h,1} f(P_{u_1}(x_0, 0, 1, x_3, O)) \geq \min_{h,1} f(P_{u_1}(x_0 - 1, 0, 1, x_3 + 1, O)) \tag{6.39}$$

Equation (6.39) reduces using (5.17a) to

$$f(x_0 - 1, 1, 1, x_3, O) \geq f(x_0 - 2, 1, 1, x_3 + 1, O)$$

which is true by (5.19b).

∘

$$\min_{1,b} f(P_{u_2}(x_0, 0, 0, 0, O)) \geq \min_{1,b} f(P_{u_2}(x_0 - 1, 0, 0, 0, O)) \tag{6.40}$$

Equation (6.40) is easily seen to be true from (5.18a). Hence (6.20a) is proved.

- **To Prove That**

$$Tf(x_0, 1, 1, 0, O) \geq Tf(x_0, 1, 1, 0, I) \tag{6.21}$$

We will consider the case $x_0 \geq 1$. Equation (6.21) will be true provided equations (6.41) and (6.42) below are satisfied.

o

$$f(P_{u_0}(x_0 - 1,1,1,0,O)) \geq f(P_{u_0}(x_0 + 1,1,1,0,O)) \qquad (6.41)$$

Equation (6.41) follows from (5.6).

o

$$\beta\mu_1 f(x_0 - 1,1,1,1,O) + \beta\mu_2 \min_{\{h,2\}} f(P_{u_2}(x_0,1,0,0,I)) =$$

$$\beta\mu_1 f(x_0 - 1,1,1,0,O) + \beta\mu_2 \min_{\{h,2\}} f(P_{u_2}(x_0,1,0,1,I)) \qquad (6.42)$$

This reduces to proving the following two equations (6.42′) and (6.42″).

$$\mu_1[f(x_0 - 1,1,1,1,O) - f(x_0 - 1,1,1,0,O)] =$$

$$\mu_2[f(x_0 - 1,1,1,1,I) - f(x_0 - 1,1,1,0,I)] \qquad (6.42')$$

and

$$\mu_1[f(x_0 - 1,1,1,1,O) - f(x_0 - 1,1,1,0,O)] =$$

$$\mu_2[f(x_0,1,0,1,I) - f(x_0,1,0,0,I)] \qquad (6.42'')$$

But equations (6.42′) and (6.42″) are true by (5.21a) and (5.21b). Hence (6.21) is verified.

6.3 Optimal Control of the Slower Server

In this section we provide proofs for Lemmas 5.7.1 and 5.7.2. By virtue of (5.17a-c), we can restrict out attention to those policies which keep the faster server busy whenever possible.

Lemma 6.3.1. *The equations (5.23a-e) in Lemma 5.7.1 propagate under the dynamic programming operator, i.e.,*

$$Tf(x_0,1,1,0,O) - Tf(x_0-1,1,1,0,O)$$

$$= Tf(x_0,1,1,0,I) - Tf(x_0-1,1,1,0,I) \qquad x_0 \geq 0 \quad (6.43a)$$

$$Tf(x_0,1,1,x_3,Z) - Tf(x_0-1,1,1,x_3,Z)$$

$$= Tf(x_0,1,1,x_3-1,Z) - Tf(x_0-1,1,1,x_3-1,Z) \qquad x_0,x_3 \geq 1 \quad (6.43b)$$

$$Tf(x_0,1,0,x_3,I) - Tf(x_0-1,1,0,x_3',I)$$

$$= Tf(y_0,1,0,y_3,I) - Tf(y_0-1,1,0,y_3',I)$$

$$x_0,y_0,x_3,x_3',y_3,y_3' \geq 0, x_3-x_3' = y_3-y_3' \quad (6.43c)$$

$$Tf(x_0,1,1,x_3,I) - Tf(x_0,1,1,x_3',I)$$

$$= Tf(y_0,1,0,y_3,I) - Tf(y_0,1,0,y_3',I)$$

$$x_0,y_0,x_3,x_3',y_3,y_3' \geq 0, x_3-x_3' = y_3-y_3' \quad (6.43d)$$

$$Tf(x_0,1,1,x_3,O) - Tf(x_0,1,1,x_3',O)$$

$$= Tf(y_0,0,1,y_3,O) - Tf(y_0,0,1,y_3',O)$$

$$x_0,y_0,x_3,x_3',y_3,y_3' \geq 0, x_3-x_3' = y_3-y_3' \quad (6.43e)$$

Proof. We provide a proof for (6.43a). The proofs of (6.47b-d), being quite similar, are left to the interested reader.

- **To Prove That**

$$Tf(x_0,1,1,0,O) - Tf(x_0-1,1,1,0,O)$$

$$= Tf(x_0,1,1,0,I) - Tf(x_0-1,1,1,0,I)$$

Equation (6.43a) will be satisfied provided the following equations (6.44)-(6.46) are true.

o

$$f(x_0+1,1,1,0,O) - f(x_0,1,1,0,O) = f(x_0+1,1,1,0,I) - f(x_0,1,1,0,I) \tag{6.44}$$

Equation (6.44) follows by (5.23a).

o

$$\begin{aligned} &\min_{h,1} f(x_0,0,1,1,O) - \min_{h,1} f(x_0-1,0,1,1,O) \\ &= \min_{h,1} f(x_0,0,1,0,O) - \min_{h,1} f(x_0-1,0,1,0,O) \end{aligned} \tag{6.45}$$

Equation (6.45) follows by (5.23b) and (5.23e).

o

$$\begin{aligned} &\min_{h,2} f(x_0,1,0,0,I) - \min_{h,2} f(x_0-1,1,0,0,I) \\ &= \min_{h,2} f(x_0,1,0,1,I) - \min_{h,2} f(x_0-1,1,0,1,I) \end{aligned} \tag{6.46}$$

Equation (6.46) follows by (5.23b-d).

∎

We now state the proof of Lemma 5.7.2, which is basically the one given by Maglaris [50], except for the fact that now it is necassary for (5.23a) to hold in order that (5.24a-d) propagate.

Lemma 6.3.2. *Equations (5.24a-d) are propagated under the the dynamic programming operator, i.e.,*

$$T\Delta_n(i) \le T\tilde{\Delta}_n(i) \qquad i \ge 2 \tag{6.47a}$$

$$T\Delta_n(i) \ge T\Delta_n(i-1) \ge 0 \qquad i \ge 2 \tag{6.47b}$$

$$T\tilde{\Delta}_n(i) \ge T\tilde{\Delta}_n(i-1) \ge 0 \qquad i \ge 2 \tag{6.47c}$$

$$T\Delta_n(i) \ge T\tilde{\Delta}_n(i-1) \qquad i \ge 2 \tag{6.47d}$$

Proof. At step zero initialize $f_O(i-1,1,0,0,I) and f_O(i-2,1,1,0,0,I)$ to arbitrary positive values such that equations (5.24a-d) hold , and $f_O(i-2,1,1,0,0,I) \geq f_O(i-1,1,0,0,I)$ for $i \leq K'$ and $f_O(i-2,1,1,0,0) \leq f_O(i-1,1,0,0,I), i > K'$. for a threshold K'. A possible choice as given in [50] is.

$$f_O(i-1,1,0,0,I) = i \qquad i \leq K' \tag{6.48a}$$

$$f_O(i-2,1,1,0,I) = i+1 \qquad i \leq K'+1 \tag{6.48b}$$

$$f_O(K',1,1,0,I) = f_O(K'-1,1,1,0,I) + 2 = K'+3 \tag{6.48c}$$

$$f_O(K'+m-2,1,1,0,I) = f_O(K'+m-3,1,1,0,I) + m, \qquad m > 2 \tag{6.48d}$$

$$f_O^\beta(j-1,1,0,0,I) = f_O^\beta(j-2,1,1,0,O) \qquad j \geq K'+1 \tag{6.48e}$$

from which $\Delta(i) = \tilde{\Delta}(i), i \leq K', \Delta(K'+1) = 1$ and $\tilde{\Delta}(K'+1) = 2, \Delta(K'+m) = \tilde{\Delta}(K'+m) = m$.

At the nth step we assume that (5.24a-d) hold and the sequence f_n defines a threshold at some value K,

$$f_n(K-1,1,0,0,I) < f_n(K-2,1,1,0,I), \quad f_n(K,1,0,0,I) \geq f_n(K-1,1,1,0,I) \tag{6.49}$$

From (6.49) it follows that

$$\Delta_n(K+1) \leq f_n(K-1,1,1,0,I) - f_n(K-1,1,0,0,I) \leq \Delta(K+1,a) \tag{6.50}$$

and using the dynamic programming operator we can write down the following equations,

$$T\tilde{\Delta}_n(1) = 1 + \beta[\lambda\tilde{\Delta}_n(2) + \mu_2\tilde{\Delta}(1)] \tag{6.51a}$$

$$T\tilde{\Delta}_n(i) = 1 + \beta[\lambda\tilde{\Delta}_n(i+1) + \mu_1\tilde{\Delta}_n(i-1) + \mu_2\tilde{\Delta}_n(i)] \qquad 2 \leq i \leq K \tag{6.51b}$$

$$\begin{aligned} T\tilde{\Delta}_n(K) = 1 + \beta[\lambda[f_n(K-1,1,1,0,I) - f_n(K-1,1,0,0,I)] + \mu_1\tilde{\Delta}_n(K-1) \\ + \mu_2\tilde{\Delta}_n(K)] \end{aligned} \tag{6.51c}$$

$$T\tilde{\Delta}_n(K+1) = 1 + \beta[\lambda\Delta_n(K+2) + \mu_1\tilde{\Delta}_n(K)$$

$$+\mu_2[f_n(K-1,1,1,0,I) - \text{[illegible]} K-1,1,0,0,I)]] \qquad (6.51d)$$

$$T\tilde{\Delta}_n(K+2) = 1+\beta[\lambda\Delta_n(K+3) + \text{[illegible]} 1,1,1,0,I) - f_n(K-1,1,0,0,I)]$$

$$\text{[illegible]} \qquad (6.51e)$$

$$T\tilde{\Delta}_n(j) = 1+\beta[\lambda\Delta_n(j+1) + \text{[illegible]} \; \mu_2\Delta_n(j)] \qquad j > K+2 \quad (6.51f)$$

$$T\Delta_n(2) = \text{[illegible]} + \mu_2\tilde{\Delta}_n(1)] \qquad (6.52a)$$

$$T\Delta_n(i) = 1+\beta[\lambda\Delta_n(i+1)+\mu_1 \text{[illegible]}(i-1)] \qquad 2 < i < K+2, \quad (6.52b)$$

$$T\Delta_n(K+2) = 1 \text{[illegible]} 3) + \mu_1\Delta_n(K+1)$$

$$+\mu_2[f_n(K-1, \text{[illegible]} K-1,1,0,0,I)] \qquad (6.52c)$$

$$T\Delta_n(j) = 1+\beta[\lambda\Delta_n(j+1)+\mu_1 \text{[illegible]} \Delta_n(j-1)] \qquad j > K+2, \quad (6.52d)$$

We next show that equations [illegible] satisfied by $\Delta_{n+1} = T\Delta_n$.

First consider equations (6.51a [illegible] quations (5.24a-d) and (6.50) it can be very easily shown that $\tilde{\Delta}_{n+1}(i)$ [illegible] and non deceasing on i. Similarly from (6.52a-d), $\Delta_{n+1}(i)$ is positive [illegible] creasing. Comparing (6.52a) with (6.51a), we have that $\Delta_{n+1} \geq \tilde{\Delta}_n$ [illegible] (6.52b) and (6.51b), $\Delta_{n+1}(i) \geq \tilde{\Delta}_{n+1}(i-1), i \leq K$ and with identical [illegible] $_1(i) \geq \tilde{\Delta}_{n+1}(i-1)$ for all i. Finally by using similar term by term comparis[illegible] conclude that $\Delta_{n+1}(i) \leq \tilde{\Delta}_{n+1}(i)$. From this it follows that if there exists an integer $K' \geq 1$ such that

$$T\tilde{\Delta}(K') \text{[illegible]} K')$$

and

$$T\tilde{\Delta}(K'+1) \geq T\Delta(K'+1)$$

then K' will be a threshold for the next iteration.

APPENDIX A

Some useful properties of the convex increasing stochastic ordering are stated in this appendix without proof. For full proofs, the reader may consult the monographs [63] or [72].

The RV $X = (X_1, \ldots, X_n)$ *is smaller than the RV* $Y = (Y_1, \ldots, Y_n)$ *in the convex increasing sense (denoted by* $X \leq_{ci} Y$*), if*

$$E[f(X)] \leq E[f(Y)]$$

for all integrable convex increasing functions $f : \mathbb{R}^n \to \mathbb{R}$.

(1): [Corollary 1.3.1 a., p. 1(, [72]] For non-negative RV's X and Y with $X \leq_{ci} Y$,

$$E[X^r] \leq E[Y^r] \qquad r = 1, 2 \ldots (A1)$$

whenever the expectations exist.

More generally, for RV's X and Y with $E[X] = E[Y]$ and $X \leq_{ci} Y$,

$$E[X^r] \leq E[Y^r] \qquad r = 2, 4, \ldots (A2)$$

(2): [Corollary 8.5.2, p. 271, [63]] If X and Y are non-negative RV's such that $E[X] = E[Y]$, then $X \geq_{ci} Y$ if and only if

$$E[h(X)] \geq E[h(Y)] \qquad (A3)$$

for all *convex* $h : \mathbb{R} \to \mathbb{R}$.

(3): [Corollary 8.5.3, p. 272, [63]] If X and Y are non-negative RV's with $E[X] = E[Y]$, then

$$X \leq_{ci} Y \quad \text{iff} \quad -X \leq_{ci} -Y \qquad (A4)$$

(4): [Proposition 8.5.4, p. 272, [63]] If [illegible]$,\ldots,X_n)$ are independent and $(Y_1,\ldots,Y_n)$ are independent, and $X_i \le$ [illegible] Y_i for $i = 1,\ldots,n$, then

$$g(X_1,\ldots \text{[illegible]} \le \text{[illegible]} (Y_1,\ldots,Y_n) \tag{A5}$$

for all non-decreasing convex [illegible] $I\!R^n \to I\!R$.

A non-negative RV X is said to [illegible] ***r than used in expectation*** (NBUE) if

$$E[X - \text{[illegible]} E[X] \tag{A6}$$

for all $a \geq 0$.

(5): [Proposition 8.6.1, p. 273, [63] [illegible] BUE distribution having mean μ, then

$$\text{[illegible]}\, \mu) \tag{A7}$$

where $\exp(\mu)$ is the exponential [illegible] on with mean μ.

APPENDIX B

Some useul properties of the strong stochastic ordering are stated in this appendix without proof. For proofs the reader may consult [63] or [72].

A RV X is stochastically smaller (or smaller in distribution) than a RV Y, denoted by $X \leq_{st} Y$, equivalently their respective distribution functions F and G satisfy $F \leq_{st} G$, if for all $x \in \mathbb{R}$

$$F(x) \geq G(x) \tag{B1}$$

(1): [Theorem 1.2.2, p. 5, [72]] The inequality

$$\int_0^\infty f(t)dF_1(t) \leq \int_0^\infty f(t)dF_2(t) \tag{B2}$$

holds for all integrable non-decreasing functions f, if and only if $F_1 \leq_{st} F_2$.

For given f, (B2) holds for all F_1, F_2 with $F_1 \leq_{st} F_2$ only if f is non-decreasing.

(2): *Lehmanns Theorem* [Proposition 1.2.1, p.4, [72]] F and G are distributions such that $F \leq_{st} G$, if and only if there exist RV's X and Y defined on the same probability space $(\Omega, \mathbb{F}, P)$ for which

$$X(\omega) \leq Y(\omega) \qquad \text{for all } \omega \in \Omega \tag{B3}$$

and

$$\begin{aligned} P(\{\omega : X(\omega) \leq x\}) &= F(x) \\ P(\{\omega : Y(\omega) \leq y\}) &= G(y) \end{aligned} \tag{B4}$$

(3): [Example 8.2(a), p.256, [63]]. If $(X_1,\ldots,X_n)$ are independent and $Y_1,\ldots,Y_n)$ are independent, and $X_i \leq_{st} Y_i, 1 \leq i \leq n$, then for any non-decreasing integrable function $f : \mathbb{R}^n \to \mathbb{R}$,

$$f(X_1,\ldots,X_n) \leq f(Y_1,\ldots,Y_n) \tag{B5}$$

(4): *The weak convergence property.* [Proposition 1.2.3, p. 6, [72]].

Assume that the sequences $\{F_n\}_1^\infty$ and $\{G_n\}_1\infty$ converge weakly to F and G and that $F_n \leq_{st} G_n$, then

$$F \leq_{st} G \tag{B6}$$

A non-negative RV is defined to be *new better than used* (NBU) if

$$P[X - a \leq x \mid X \geq a] \leq P[X \geq x] \tag{B7}$$

for all $a \geq 0$.

(5): [Proposition 1.6.2, p. 19, [72]]. If F is a NBU distribution having mean μ, then

$$F \leq_{st} \exp(\mu) \tag{B8}$$

REFERENCES

[1] T.S. Axelrod, "Effects of synchronization barriers on multiprocessor perfomance," *Parallel Computing*, Vol. 3, pp. 129-140, (1986).

[2] F. Baccelli, "Two parallel queues created by arrivals with two demands: The $M/G/2$ symmetrical case," *Report de Recherche*, No. 426 INRIA, (1985).

[3] F. Baccelli, "Queues with locking, the multiclass problem," *Teletraffic Analysis and Computer Perfomance Evaluation*, Elsevier Science Publishers B.V. (North Holland), (1986).

[4] F. Baccelli, C.A. Courcoubetis and M.I. Reiman, "Construction of the stationary regime of queues with locking," AT&T Bell Laboratories, Technical Memorandum.

[5] F. Baccelli, E. Gelenbe and B. Plateau, "An end-to-end approach to the resequencing problem," *JACM*, Vol. 31, No. 3, pp. 474-485, (1984).

[6] F. Baccelli and A. Makowski, "Simple computable bounds for the Fork-Join queue," Proceedings of the Johns Hopkins Conference on Information Sciences and Systems, The John Hopkins University (1985).

[7] F. Baccelli, A. Makowski and A. Schwartz, "Fork-Join queue and related systems with synchronization constraints: Stochastic ordering, approximations and computable bounds," Electrical Engineering Technical Report, No. TR-87-01 University of Maryland, College Park (1987).

[8] F. Baccelli and W. A. Massey, "Series-parallel, fork-join queueing networks and their stochastic ordering," AT&T Bell Laboratories Memorandum, (1985).

[9] F. Baccelli and Ph. Robert, "Analysis of update response times in a distributed data base maintained by the time stamp ordering lgorithm," *PERFORMANCE'83* (1983).

[10] F. Baccelli and T. Znati, "Queueing algorithms with breakdowns in data base

modelling," *Rapport de Recherche*, No. 50, INRIA, Le Chesnay, France (1980).

[11] R. Barlow and F. Proschan, *Statistical Theory of Reliability and Life Testing*, Holt, Rinehart and Winston, (1975).

[12] F. Baskett, K.M. Chandy, R.R. Muntz and F.G. Palacios, "Open, closed and mixed networks of queues with different classes of customers," *JACM* Vol. 22, pp. 248-260 (1975).

[13] M. Ben-Ari, *Principles of Concurrent Programming*, Prentice Hall International (1982).

[14] A. Bernstein and M. Goodman, "Concurrency control in distributed data base systems," *ACM Computing Surveys*, Vol. 13, No. 2, (1981).

[15] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall (1987).

[16] G. Broomell and J.R. Heath, "Classification categories and historical development of circuit switching topologies," *Comp. Surveys*, Vol. 15, No. 2, pp. 95-132 (1983).

[17] K. Bharat-Kumar and P. Kermani,"Analysis of a resequencing problem in communications," *Proc. IEEE INFOCOM*, pp. 303-310 (1983).

[18] R. L. Disney and D. König, "Queueing networks: A survey of their random processes," *SIAM Review*, Vol. 27, No. 3, pp. 335-403 (1985).

[19] J. M. Feuvre, "Queueing analysis of a reordering system," *PERFORMANCE'81*, (1981).

[20] L. Flatto and S. Hahn, "Two parallel queues created by arrivals with two demanda I," *SIAM J. Appl. Math.* Vol. 44, pp. 1041-1053 (1984).

[21] E. Gelenbe and K. C. Sevcik, "Analysis of update synchronization for multiple copy data bases," *Proc. of the 3rd Berkeley Workshop on Distributed Data Management and Computer Networks*, National Technical Information Service, Springfield, Va. (1978).

[22] B. Hajek, "Optimal control of two interacting service stations," *IEEE Trans. on Auto. Control*, Vol. 29, pp. 491-499 (1984).

[23] G. Harrus and B. Plateau, "Queueing analysis of a re-ordering issue," *PERFORMANCE'81*, pp. 251-269 (1981).

[24] P. Heidelberger and S.K. Trivedi, " Queueing network models for parallel processing with asynchronous tasks," *IEEE Trans. on Computers* Vol. 31, No. 11, pp. 1099-1109 (1982).

[25] P. Heidelberger and S.K. Trivedi, "Analytic queueing models for programs with internal concurrency," *IEEE Trans. on Computers* Vol. 32, No. 1, pp. 73-82 (1983).

[26] C.A.R. Hoare, *"Communicating sequential processes,"* Prentice- Hall International, London, (1985).

[27] C.A.R. Hoare, "Monitors: An operating system structuring concept," *CACM*, Vol. 17, No. 10, pp. 549-557 (1974).

[28] E. Horlatt and D. Mailles, "Etude du resequencement dans un reseau de files d'attente," Manuscript, Universite Pierre et Marie Curie, No. 125, (1986).

[29] D.R. Jacobs and S. Schach, "Stochastic order relationships between $GI/G/K$ systems," *Ann. Math. Statist.*, Vol. 43, no. 5, pp. 1623-1633 (1972).

[30] A. Jean-Marie, "Load balancing in multi-stage interconnection networks," Manuscript, INRIA, (1986).

[31] J. Kiefer and J. Wolfowitz, "On the theory of queues with many servers," *Trans. Amer. Math. Soc.*, No. 78, pp. 1-18 (1955).

[32] F. Kamoun, L. Kleinrock and R. Muntz, "Queueing analysis of the ordering issue in a distributed database concurrency control mechanism," *Proceedings of the 2nd International Conference on Distributed Computing Systems* (1981).

[33] H.R. Kanakia and F.A. Tobagi, "On distributed computations with limited resources," *IEEE Trans. on Comp.*, vol. 36, no. 5, pp. 517-528 (1987).

[34] S. Karlin and H.M. Taylor, *A First Course in Stochastic Processes*, Academic Press (1975).

[35] S. Karlin and H.M. Taylor, *A Second Course in Stochastic Processes*, Academic Press (1981).

[36] F.P. Kelly, *Reversibility and Stochastic Networks*, John Wiley & Sons (1979).

[37] J.F.C. Kingman, "The effect of queue discipline on waiting time variance,"

Proc. Camb. Phil Soc, No. 58, pp. 163-164 (1962).

[38] L. Kleinrock, "Distributed Systems," *CACM* Vol. 28, No. 11, pp. 1200-1213 (1985).

[39] L. Kleinrock, *Queueing Systems* Vol. 1 & 2, John Wiley 1975.

[40] C.P. Kruskal and A. Weiss, "Allocating independent subtasks on parallel processors," *IEEE Trans. on Soft. Engineering* , Vol. 11, pp. 1001-1016, (1985).

[41] P.R. Kumar and W. Lin, "Optimal control of a queueing system with two heterogenous servers," *IEEE Trans. on Auto. Control*, vol. 28, pp. 696-703 (1984).

[42] P.R. Kumar, P. Varaiya, *Stochastic Systems*, Prentice-Hall Inc. (1987).

[43] L. Lamport, "Time, Clocks, and the ordering of events in a distributed system," *CACM*, Vol. 21, No. 7, pp. 558-565 (1978).

[44] E.L. Lehmann, "Ordered families of distributions," *Ann. Math. Statist.*, Vol. 26, pp. 399-419 (1955).

[45] G. Le Lann, "Algorithms for distributed data sharing systems which uses tickets," *Proceedings of the 3rd Berkeley Workshop on Distributed Data Management and Computer Networks*, National Technical Information Service, Springfield, Va. (1978).

[46] S.A. Lippman, "Applying a new device in the optimization of exponential queueing systems," *Oper. Res.*, Vol. 23, pp. 687-710 (1975).

[47] S.A. Lippman, "Semi-Markov decision processes with unbounded rewards," *Manag. Sci.*, Vol. 19, pp. 717-731 (1973).

[48] R. M. Loynes, "The stability of a queue with non-independent interarrival and service times," *Proc. Cambridge Philo. Soc.* Vol. 5, pp. 497-520 (1962).

[49] Y.C. Luke Lien, "Evaluation of the resequencing delay in a Poisson queueing system with two heterogeneous servers," IBM Research Centre Report, Yorktown Heights, (1985).

[50] B.S. Maglaris, "An optimal local policy for two-level adaptive routing in computer networks," *IEEE INFOCOM'84*, pp. 284-290 (1984).

[51] D. Mitra, "Randomized parallel communications on an extension of the omega

network," AT&T Tech. Rep. (1986).

[52] D. Mitra, " Probabilistic models and asymptotic results for concurrent processes with exclusive and non-exclusive locks," *SIAM J. Comput*, Vol. 14, No. 4, pp. 1030-1051, (1985).

[53] D. Mitra and P.J. Weinberger, "Some results on database locking: Solutions, computational algorithms and asymptotics," *Mathematical Computer Perfomance and Reliability*, Proc. of the International Workshop , Pisa (1983).

[54] R. Nelson and A.N. Tantawi, "Approximate analysis of fork-join synchronization in parallel queues," *IBM Report RC11481* (1985).

[55] R. Nelson, D. Towsley and A.N. Tantawi, "Perfomance analysis of parallel processing systems," to be presented at *SIGMETRICS'87* (1987).

[56] M. F. Neuts, *Matrix-geometric solutions in stochastic models - An algorithmic approach*, The John Hopkins University Press (1981).

[57] M. F. Neuts, "Markov chains with applications in queueing theory, which have a matrix-geometric invariant probability vector," *Adv. Appl Prob*, 10, pp. 185-212 (1978).

[58] S.C. Niu, "On the comparison of waiting times in tandem queues," *J. Appl. Prob.*, Vol. 18, pp. 707-714, (1981).

[59] G.L. O'Brien, "Inequalities for queues with dependent interarrival and service times," *J. Appl. Prob.,* 12, pp. 653-656 (1975).

[60] G.L. O'Brein, "The comparison methods for stochastic processes," *Ann. Prob.*, Vol. 3, pp. 80-88 (1975).

[61] C.A. Petri, "Modelling as a communication discipline," *Measuring, Modelling and Evaluating Computer Systems*, North Holland Publishing Co. (1977).

[62] C.G. Rommel, D. Towsley, J. A. Stankovic, "Analysis of fork-join jobs using processor sharing," Tech. Rep. Univ. of Mass., Amherst (1987).

[63] S.M. Ross *Stochastic Processes*, J. Wiley & Sons, (1983).

[64] S.M. Ross *Stochastic Dynamic Programming*, Academic Press, (1983).

[65] Z. Rosberg and N. Shacham, "Resequencing delay and buffer occupancy under the selective repeat ARQ," Manuscript (1986).

[66] Z. Rosberg, P. Varaiya and J.C. Walrand, "Optimal control of service in tandem queues," *IEEE Trans. on Auto. Control*, Vol. 27, No. 3, pp. 600-610 (1982).

[67] C.L. Seitz, "The cosmic cube," *CACM*, Vol. 28, No. 1, pp. 22-33 (1985).

[68] R. Serfozo, "An equivalence between discrete and continous time Markov decision processes," *Op. Res.*, Vol. 27, pp. 616-620 (1979).

[69] S. Stidham Jr., "Optimal control of admission to queueing system," *IEEE Trans. on Auto. Control*, Vol. 30, No. 8, pp. 705-713 (1985).

[70] H.S. Stone, *High Perfomance Computer Architecture*, Addison-Wesley Publ. Co. (1987).

[71] D. Stoyan, "Bounds and approximations in queueing through monotonocity and continuity," *Op. Res.* Vol. 25, No. 5, pp. 851-863 (1977).

[72] D. Stoyan, *Comparison Methods for Queues and Other Stochastic Models*, English Translation (D. J. Daley Editor), J. Wiley & Sons, New York (1984).

[73] L. Takacs, *Introduction to the Theory of Queues*, Oxford University Press (1962).

[74] A. Thomasian and P.F. Bay, "Analytic queueing network models for parallel processing of task systems," *IEEE Trans. on Computers*, Vol. 35, No. 12, pp. 1045-1054 (1986).

[75] D. Towsley and S.P. Yu, "Bounds for two server fork-join queueing systems," submitted to Operations Research.

[76] J.D. Ullman, *Principles of Data Base Systems*, Computer Science Press (1984).

[77] J. Walrand, "A note on optimal control of a queueing system with two heterogenous servers," *Sys. and Contr. Letters*, Vol. 4, pp. 131-134 (1984).

[78] W. Whitt, "Minimizing delays in the $GI/G/1$ queue," *Op. Res.* Vol. 32, pp. 41-51 (1984).

[79] R.W. Wolff, "An upper bound for multi-channel queues," *J. Appl. Prob.*, Vol. 14, pp. 884-888 (1977).

[80] T. P. Yum and T. Ngai, "Resequencing of messages in communication net-

works," *IEEE Transactions on Communications*, Vol. Com-34, No. 2, pp. 143-149, (1986).

CURRICULUM VITAE

Name: Subir Varma

Permanent address: 6002 Springhill Drive ♯104, Greenbelt, MD, 20770.

Degree and date to be conferred: M.S.,1987.

Date of birth: July 15, 1964.

Place of birth: Patna, India.

Secondary education: Methodist High School, Kanpur, India, 1981.

Collegiate institutions attended:

- 1985-1987: University of Maryland
College Park, Maryland
M.S. in Electrical Engineering, 1987.
- 1981-1985: Indian Institute of Technology,
Kanpur, India,
B.Tech in Electrical Engineering, 1985.

Major: Communications.

Professional publications:

- "Some problems in queueing systems with resequencing," *M.S. Thesis*, Electrical Engineering Department, University of Maryland, College Park, August 1987.

Professional positions held:

- Fall 1985– : Systems Research Centre Fellow,
Electrical Engineering Department,
University of Maryland,
College Park MD 20742.