THE SCHEDULING OF QUEUES WITH NON-LINEAR HOLDING COSTS

Marilyn O'Keeffe

Thesis submitted for the degree of

Doctor of Philosophy

UNIVERSITY OF
NEWCASTLE

School of Mathematics and Statistics

University Of Newcastle Upon Tyne

Newcastle Upon Tyne

United Kingdom

November 2003

Abstract

We consider multi-class, single server queueing systems and we seek to devise policies for server allocation which minimise some long-term cost function. In most of the work to date on the optimal dynamic control of such systems, holding cost rates are assumed to be linear in the number of customers present. Such assumptions have been argued to be unrealistic and thus inappropriate: see Van Meighem (1995).

With pure priority policies, which often emerge from analyses based on linear holding cost assumptions, there is often the problem that service offered to lower priority traffic is unacceptably poor. Seeking to address such problems, we first investigate the performance of policies based on linear switching curves in an M/M/1 model with two customer types, imposing various constraints on the second moments of queue lengths. We then develop an index heuristic for a multi-class M/M/1 model with increasing convex holding cost rates. Following work by Whittle (1988), we develop the required indices and in a numerical study of two and three class systems, demonstrate the strong performance of these index policies. Performance of policies throughout the thesis, as measured by lowest costs achievable under a given policy class, (i.e. best linear switching, best threshold, or index policy) is compared with a lower bound on the minimum cost achievable under any policy. This lower bound is obtained by adopting the achievable region approach, see Bertsimas, Paschalidis & Tsitsiklis (1994) and Bertsimas & Niño-Mora (1996) in which we construct a set of constraints satisfied by the first and second moments of the queue lengths. These constraints define a relaxation of the set of achievable region performance vectors of the system. Optimisation over this relaxed region yields the lower bound. Numerical results indicate the strong performance of the index policy.

Acknowledgements

My thanks are due to Prof. Kevin Glazebrook for giving me the opportunity to take part in this research, his guidance throughout it and his advice on the completion of this thesis. I wish to thank Dr. Phil Ansell for his patience during my early programming and his supervision during the writing of the thesis. I have also appreciated the support and friendship of staff and postgraduate students in the School of Mathematics and Statistics at the University of Newcastle upon Tyne. Sincere thanks to Kathy Pike, lecturer in mathematics at South Tyneside College, who gave me the confidence to succeed. I also acknowledge with thanks the Engineering and Physical Sciences Research Council for funding this research.

Finally, I would like to thank my husband, George, and our children, John, Helen, Aidan and Stephen, for their unwavering love, support and encouragement. Bless you all.

Deo gratias.

For my parents, Harry and Winifred Smith, with love.

'But many that are first will be last, and the last first.'

Matthew 19:30

Contents

1 Introduction **7**

1.1 Markov Decision Processes 8

1.1.1 The Multi-armed bandit problem 9

1.1.2 The Gittins Index 10

1.2 More complex bandit problems 10

1.2.1 The Whittle Index 11

1.3 Dynamic Programming 13

1.4 The achievable region approach. 14

1.4.1 Strong conservation laws. 16

1.4.2 Generalised conservation laws. 20

1.5 Non-linear holding costs 23

1.6 Thesis Outline 24

2 Application of policies based on linear switching curves to problems with non-linear constraints **26**

2.1 Introduction 26

2.2 Analytical methods 29

2.2.1 The Power Series Algorithm 29

2.2.2 Enlarging the radius of convergence 31

2.2.3 The epsilon algorithm 33

2.3 Admissible Service Policies 34

2.4 Policies based on linear switching curves 37

2.5 Performance analysis of policies based on linear switching curves using the power-series algorithm 37

2.6 Bounding sets 42

2.6.1 Higher order interactions 44

2.7 A relaxation of the achievable region in a two customer class system 48

2.7.1 Optimisation over the achievable region. 54

2.8 Semidefinite programming 56

2.9 Problems involving constraints on the second moments of queue lengths . . 61

2.9.1 Computations 62

2.10 Initial Results 65

2.10.1 Single constraint on the second moment of the length of the lower priority queue 65

2.10.2 Constraints on both second moments. 66

2.11 Constraining the problem by a linear sum of the second moments. 66

2.11.1 Problems with varying ρ 67

2.11.2 Problems where $\mu_1 = \mu_2 = 1$ 69

2.11.3 Problems where $\mu_1 = 3, \mu_2 = 12$ 70

2.12 Conclusion 72

3 Whittle index Policies **73**

3.1 Introduction 73

3.2 The Model 74

3.3 Indexability and Whittle indices 77

3.3.1 Active action in n 81

3.3.2 Passive action in n 83

3.4 The Index 84

3.5 PCL-Indexability 93

4 **An Evaluation of a Whittle index policy in two simple cases with average costs.** **95**

4.1 The form of the index for a quadratic cost function 96

4.2 Whittle index policy for a two customer class problem 98

4.2.1 A specific linear switching curve 99

4.3 Calculation of C^{OPT} via the value-iteration algorithm 99

4.4 Two class problem; results 101

4.5 Whittle index policy for the three class problem 105

4.6 Approximation of the Achievable Region in a three class system. 112

4.7 The semidefinite constraints on the three class problem. 121

4.8 Calculation of C^{OPT} for the three class problem 126

4.9 Numerical examples of three class type problems 128

4.9.1 Further investigations 134

4.10 Conclusions and future work. 140

Chapter 1

Introduction

The problems of scheduling are familiar to most of us in our everyday lives. A busy NHS hospital, for example, must work within the constraints of a limited budget to balance the varying demands made on its services by patients with differing needs of treatment. The decisions made by a hospital trust at a particular time affect the choices open to it in the future and their aim is usually to maximise some targets of patient throughput or satisfaction or to minimise some aspect such as patient waiting times. So it is in the world of telecommunications and computing. The degree to which available service/processor time outstrips the demand for service means that the various types of jobs arriving for service are often forced to wait in the system until they can be processed by the server. Such a situation means it is necessary to develop policies or rules as to when a job arriving for service is served. One such policy would be simply *first come, first served* but consideration, for example, of the queue in an accident and emergency department would clearly indicate that it is not always desirable or possible to allocate service on such a simple basis. Obviously, there are patients arriving who have differing needs, and differing priorities need to be devised for them as the cost of delaying a patient with a cut finger is not as great as the cost of delaying a patient with a heart attack.

Queueing theory can assist us in addressing the real world problems of server allocation to jobs/customers of differing priorities. Such problems can sometimes be modelled by the use of Markov decision processes in the framework of stochastic optimisation. We seek to devise policies for server allocation which will minimise some long-term holding cost rate.

Ideally, service policies should be both simple to devise and implement. Such are the complexities of server allocation problems, however, that this has rarely been possible. The so-called curse of dimensionality constantly imposes limits on the computational viability of problems and related solution methods. Early work by Gittins (1979) demonstrated that for certain simple stochastic resource allocation problems, an index, $v_i(x_i)$ could be calculated for each job type. This index was simply a function of a job's type, i and its state, x_i and thus problems of dimensionality were reduced. The optimal policy under given conditions was for the server to operate on the job type with the highest index: hence the term *index policy*. Clearly the development of index based policies was highly desirable. The work presented herein represents a small part of ongoing research to extend to more complex models, the systematic design of heuristic service policies whose performance is close to optimal.

1.1 Markov Decision Processes

A Markov process is a model of a system which passes through a succession of states, each determined by a succession of transition probability distributions which in turn depend upon the current state of the system. We add to this, firstly, a set of decisions at each stage, and on which the probability distribution governing the next stage of the system depends, and, secondly, a set of possible rewards at each stage, depending on the decision and the subsequent transition made. Thus, we define a discrete time Markov Decision Process (MDP) as follows: A dynamic system which has decision epochs at equidistant points of time say, $t = 0, 1, ...$

- A state space, I
- For each state $i \in I$, a set of possible actions, $A(i)$

- A real valued reward (or cost) system, $R(i, a)$, depending on which action, a, is taken in state i.

We also assume the Markov property, i.e. the effects of making a decision in a given state depend only on that state and not on any past states.

MDPs serve as useful models for a wide variety of systems where sequential decisions are to be made. Blackwell (1965) considered discounted MDPs, first those with a discrete time process and then extended his work to include those processes where the intervals between successive decision times are random variables. The latter are called semi-Markov decision processes. Our work concerns mainly average cost problems but we also consider a problem with an infinite time horizon MDP in which future rewards (or costs) are discounted, so as to ensure that the total reward obtained or cost incurred is finite. Our aim is to find policies which either maximise this reward function or minimise some holding-cost function.

Some of the earliest work on such problems was in the context of queueing systems. Among the more prominent of the early contributors were Cobham and Harrison, both of whom considered queues with non-pre-emptive priorities. Cobham (1954) argued that as the load on a system increases, then the need to prioritise the jobs in some way increases and he recommended that as jobs of the highest priority would delay those of lower priorities, the frequency of jobs being assigned as high-priority should be kept as low as possible and that the servicing times of those high-priority jobs should be as short as possible. Harrison (1975) showed for his model that a policy whereby the jobs are serviced according to a strict priority rule is optimal.

1.1.1 The Multi-armed bandit problem

Multi-armed bandit problems are a type of MDP which involve the dynamic allocation of some limited resource to a fixed collection of competing projects. The aim is to maximise the total expected rewards or minimise the total expected costs of delay. A simple example of such a problem would be one involving m drilling machines which can be used to prospect for oil at n different locations and $m < n$. It is important to note that at those locations where no drilling takes place at a given moment in time, it is assumed that the states of those locations remain unchanged.

1.1.2 The Gittins Index

Gittins & Jones (1974) produced the first solution of one of the multi-armed bandit problems. Proof of optimality uses an interchange argument and the result takes the form of an index, the Gittins index, a function of the project type and state. The optimal policy is simply, at each decision epoch, to serve that customer type with the largest index value. When the system modelled is that of a single server queue with K customer types and costs are linear in each class queue length, and the goal is to minimise the long run holding cost rate, then the optimal policy under given conditions is to schedule jobs in decreasing order of $c_k\mu_k$, where c_k is the cost incurred per unit of time until the job is completed and μ_k is the rate at which customers of type k are served. (See, for example Coffman & Mitrani (1980).) Such a policy has clear advantages in that calculation of the indices is extremely simple and does not involve the K-dimensional calculations of classical dynamic programming. The implementation of the policy is also quite straightforward.

Clearly there was a great incentive for work to be carried out in generalising the types of bandit problems for which optimal index solutions could be found.

1.2 More complex bandit problems

In 1980, Whittle offered a proof of the Gittins index result via a dynamic programming approach; see Whittle (1980). He also extended his results to include what Gittins (1979) refers to as bandit superprocesses. Such a process has an extra decision variable, u_i added so that at a decision time, we must decide not only the project i which is operated on but also which procedure, u_i is adopted.

At the same time as Whittle produced this work, Nash (1980), considered a problem where the reward gained for operating project i depended not simply on the state of project i but on the states of all of the other projects in a multiplicatively separable fashion. He derived an index for such a problem and proved the optimality of the corresponding index policy. Whittle (1981) considered arm acquiring bandits, one of a general class of problems now known as branching bandits. See Glazebrook (1976), Weiss (1988), Klimov (1974) and Tcha & Pliska (1977). Whittle's problem can be thought of as a multi-armed bandit which develops more arms, or projects, as time passes. Such problems are particularly useful to us here in the modelling of multi-class queueing systems but their use extends also to systems in the fields of medicine, agriculture and technology, where, over time, there are greater choices to be made because of technical advances. Again, Whittle found an optimal Gittins index solution to the problem.

Obviously, the Gittins index approach was offering optimal solutions to a range of bandit problems. The work in extending the problems to which index solutions provided such solutions continued throughout the 1980s. Most contributions analyse the models via a dynamic programming approach: see Glazebrook (1982), Glazebrook (1991) and Varaiya, Walrand & Buyukkoc (1985).

1.2.1 The Whittle Index

In ground-breaking work, it was Whittle (1988) who first suggested that index solutions could be developed for restless bandit problems. In the multi-armed bandits of Gittins (1979), the bandits are assumed to be static during their passive phases. Hence, if we have n bandits or projects from which we must choose one to be active, the states of the $n-1$ on which no action is taken remain unchanged. Whittle describes a population of n projects, or restless bandits, which continue to evolve, each according to its own set of rules, whether they are functioning (active) or not (passive) but according to distinct transition laws for active/passive. The problem, like that of our earlier drilling problem, is to choose which of the projects should be active at any given time so as to maximise the expected reward rate earned given that at each decision epoch we are allowed to operate only m, where $m < n$. He relaxed the problem to one which requires m projects to be active on average. He then solved this relaxed version using Lagrangian methods. He argued that the Lagrange multiplier associated with the constraint on total processing, takes the form of an index, $v_i(x_i)$, which simplifies to that of the Gittins index when the bandits are assumed static when passive. He asserted that for m and n large and in constant ratio, the policy whereby at any given point, the m active bandits are those with the largest current index is asymptotically optimal. It is important to note, however, that in general Whittle's proposed indices do not necessarily exist for every MDP. A given MDP must be shown to have the property of *indexability* and even when such indices do exist, they are not in general optimal.

This approach of Whittle's was heuristic and was essentially based on simplifications of the problem in the undiscounted case. Early work by Cox & Smith (1961) suggested the optimality of service policies where the server chooses from the waiting customers according to a fixed ordering of the classes and the costs to be optimised are linear. Weber & Weiss (1990) and Weber & Weiss (1991) showed, mathematically, that Whittle's conjecture of asymptotic optimality as m and n approach infinity is true if the differential equation representing the fluid approximation to the index policy has a globally stable equilibrium point. They show that, although this is not always so, exceptions are extremely rare.

The practical applications of such models are many. Bertsimas & Niño-Mora (2000) mention the following examples: clinical trials, aircraft surveillance, worker scheduling (see Whittle (1988)), police control of drug markets and control of a make-to-stock production facility. For the latter see also Veatch & Wein (1996). Thus, there is great motivation to extend the research to seek index solutions for such problems.

Many approaches will be, like that of Whittle's, heuristic in nature. The quality of such policies has usually been measured by a comparison of their empirical performance with that of a minimum cost to the problem provided by means of standard dynamic programming techniques, where this was possible. When this was not possible, usually because of dimensionality problems, assessment was made simply by a comparison of the performances of a variety of heuristic policies, usually in a simulation study, and, as such, offered little insight concerning their degree of suboptimality. A relatively new approach, that of the *achievable region* helps us to address such difficulties among others.

1.3 Dynamic Programming

The techniques known collectively as Dynamic Programming (DP) were devised for optimisation problems involving sequential decisions. Thus, they have always been considered the natural framework for the optimal solution of MDPs. Our concept of a solution to an MDP is a policy, i.e. a set of rules determining which action should be taken at each decision epoch, for each possible state.

It was Bellman (1957) who first propounded the ideas of DP and stated his *principle of optimality* concerning the optimal policy, that whatever the initial state and initial decision of a sequential decision process, the remaining decisions must also form an optimal policy. This is true for many models when future states are independent of all past states and depend only on the current state, so that the path taken to arrive at the present state is irrelevant for decision-making. We suppose that in our MDP defined in Section 1.1, we choose action a in state i, then the following occur:

- We incur an immediate cost, $c_i(a)$ or reward, $r_i(a)$
- The system will move to state j at the next decision epoch with probability $p_{ij}(a)$, where $\sum_{j\in I} p_{ij}(a) = 1$, $i \in I$.

Assume that our goal is to optimise the discounted costs/rewards to infinity. By applying the principle of optimality to such an MDP, we obtain a set of recursive equations for functions defined on state space I, the solution of which will give us the optimal cost/reward. Rarely, such equations can be solved analytically and an optimal policy deduced from them. When this is not possible, we can seek to solve the equations numerically, by iterative approaches, although the application of such methods is frequently limited by the problem of dimensionality. An n class queueing problem usually has an n dimensional state space, and thus our ability to compute solutions is curtailed by the exponentially increasing size of the problem. For all practical purposes, we are severely restricted in DP to the optimal solution of such problems with relatively small n.

1.4 The achievable region approach.

Until fairly recently, the standard technique of approach to stochastic scheduling problems was dynamic programming. Although it is true to say that many such problems can be set within such a framework, as we have already mentioned above, its effectiveness is rapidly curtailed by the size of the resulting calculations. This is especially found to be a problem, as one might expect, in more complex stochastic optimisation problems.

An alternative approach has been that of the achievable region. This approach has its roots in mathematical programming. The basic aim of the achievable region approach is to provide a general framework for the solution of stochastic scheduling problems. In general terms, it operates as follows:

- Define some suitable performance measure and characterise the performance space (the set of all possible performances of the system under all possible policies) using a set of physical laws, usually conservation laws, which describe the system.

- Solve a mathematical programming problem over the performance space.

For those problems involving the optimal scheduling of multiclass queueing systems, each scheduling policy usually has associated with it a performance vector, whose ith component is the performance measure associated with customers of type i. The achievable region $\mathcal{X}$ of a problem is the set of performance vectors, e.g. mean queue lengths, of all of the admissible policies and the solution to the associated scheduling problem may sometimes be found by solving a mathematical program with $\mathcal{X}$ as the feasible set. Thus, given a vector of performance measures $\boldsymbol{x}$, the expected number of jobs of each type in a multi-class queueing system, and a cost function $c(x)$, we would seek Z^* the minimum cost achievable under any admissible policy or $\underline{Z}$, a lower bound on it. The minimum achievable cost is

$$Z^* = \min \ c(x)$$
$$\text{subject to}$$
$$x \in \mathcal{X}$$

where $\mathcal{X}$ is the exact performance region. Ideally this region would be characterised explicitly by means of algebraic constraints. In those cases where it is not possible to characterise the exact performance region, constraints may be generated to obtain a relaxation of the achievable region, i.e. a set which contains it. See, for example (Bertsimas & Niño-Mora 2000). Let $\mathcal{P} \supseteq \mathcal{X}$ be a relaxation of the performance region, then we can obtain a lower bound on Z^* by the solution of:

$$\underline{Z} = \min \ c(x)$$
$$\text{subject to}$$
$$x \in \mathcal{P}.$$

The solution to even this relaxed program can lead to good, i.e. close to, optimal policies. In this it is clear that the generation of constraints (i.e. relations satisfied by all x) is of paramount importance as is the design of good policies from the solutions.

The characterisation of the achievable region of a stochastic scheduling problem was first achieved by Coffman & Mitrani (1980) and Gelenbe & Mitrani (1980). Their work concerned optimal control problems for multi-class M/M/1 and M/G/1 queues. They showed that by identifying every scheduling policy with a performance vector consisting of mean response times for each customer class, the set of achievable performances (i.e. those performance vectors which are the result of adopting some admissible scheduling policy) can be characterised as a region bounded by a set of linear constraints. The latter arise from the physical laws which describe the system's behaviour subject to different scheduling policies. The problem of optimising a linear function of customer class waiting times over admissible policies is reduced to a linear programming problem which is able to be solved by standard methods. We shall consider optimisation problems involving the minimisation of customer queue length but the same approach applies. Coffman & Mitrani (1980) and Gelenbe & Mitrani (1980) showed the achievable region of a multi-class M/M/1 queue under non-idling service policies to be a polyhedron and, significantly, showed that the optimality of priority-index policies derives directly from this polyhedral structure.

The work was extended with the aim of developing a general framework for the analysis of many stochastic scheduling problems. See, for example Federgruen & Groenevelt (1988) and Shanthikumar & Yao (1992). The latter pair introduced the concept of strong conservation laws and proved results concerning the form of the achievable region, once such laws are shown to hold.

1.4.1 Strong conservation laws.

Shanthikumar & Yao (1992) defined so-called *strong conservation laws* for systems and showed that, if these held, then the achievable region, $\mathcal{X}$, was of a particular form, a polymatroid, the vertices of which are the performance vectors of the absolute priority rules. In this subsection, we outline their work. Let $E = \{1, \ldots, n\}$ be a set of n different job types in a general queueing system. We assume scheduling strategies are non-idling and non-anticipative. Thus the server is always active when there are customers to be served and scheduling decisions are based only on the past history and current state of the system. We denote U as the class of admissible service policies and x_j^u as a performance measure of type j jobs under an admissible policy u. Let $\mathbf{x}^u = \left(x_j^u\right)_{j \in E}$ be the corresponding performance vector. For any permutation $\pi = \pi_1, \pi_2, \ldots, \pi_E$ of the n elements of set E, we denote $\mathbf{x}^\pi$ the performance vector for the scheduling policy which prioritises the job types according to the permutation. Thus, π_1 is given the highest priority, then π_2 and so on down to π_n being given lowest priority. We shall use the notation $x(S) := \sum_{j \in S} x(j)$, for any $S \subseteq E$.

Definition 1.1 (Strong conservation laws) *If* $\exists\ b\colon 2^E \to \mathbb{R}^+$, *a set function which satisfies the following conditions for all* $u \in U$;

$$\sum_{j \in E} x_j^u = b(E) \tag{1.1}$$

$$\sum_{j \in S} x_j^u \geq b(S), \quad all \ S \subset E \tag{1.2}$$

and such that, for all scheduling policies $\pi = \{\pi_1, \pi_2, ..., \pi_E\}$,

$$\sum_{j=1}^{r} x_{\pi_j} = b(\{\pi_1, \pi_2, ..., \pi_r\}),, \quad for \ r = 1, 2, ..., E \tag{1.3}$$

then performance vector $\mathbf{x} = (x_1, x_2, ..., x_E)$ *satisfies strong conservation laws.*

If the strong conservation laws hold, then it follows from (1.1) and (1.2) that any performance vector $\mathbf{x}$ will belong to the polyhedron, P, where

$$P = \left\{ \mathbf{x} \in \mathbb{R}_+^E : \sum_{j \in S} x_j \geq b(S), \quad all \ S \subset E; \sum_{j \in E} x_j = b(E) \right\}. \tag{1.4}$$

Shanthikumar & Yao (1992) showed that if a system satisfies strong conservation laws (Definition (1.1)) then the base function b *must* satisfy the properties listed in Definition (1.2) and the polyhedron P must be a polymatroid.

Definition 1.2 (Normalised, non-decreasing and supermodular) *Let* $b : 2^E \to \mathbb{R}^+$ *be a set function which is*

- *normalised:* $b(\varnothing) = 0$
- *non-decreasing:* $b(S) \leq b(T)$, *for all* $S \subseteq T \subseteq E$ *and*
- *supermodular:* $b(S) + b(T) \leq b(S \cup T) + b(S \cap T)$ *for all* $S, T \subseteq E$

Theorem 1.1 (Shanthikumar and Yao (1992)) *Assume the performance vector* $\mathbf{x}$ *satisfies strong conservation laws (1.1), (1.2) and (1.3) then*

- *The performance space is the polyhedron, P, described by (1.4).*

- *P is the base of a polymatroid.*

- *The performance vectors of the absolute priority rules form the vertices of P.*

This result enabled a wide range of queueing scheduling problems to be solved by simple priority rules. These included problems previously considered by Coffman & Mitrani (1980) and Federgruen & Groenevelt (1988). As an example of how the method works, we use an M/M/1 queueing system with two job types discussed by Gelenbe & Mitrani (1980). Define $V_u(t)$ as the total amount of work in the system at time t under an admissible scheduling strategy u. It is assumed that the speed of the server is 1. Figure 1.1 illustrates a typical realisation of $V_u(t)$. Whenever a customer arrives, $V_u(t)$ jumps

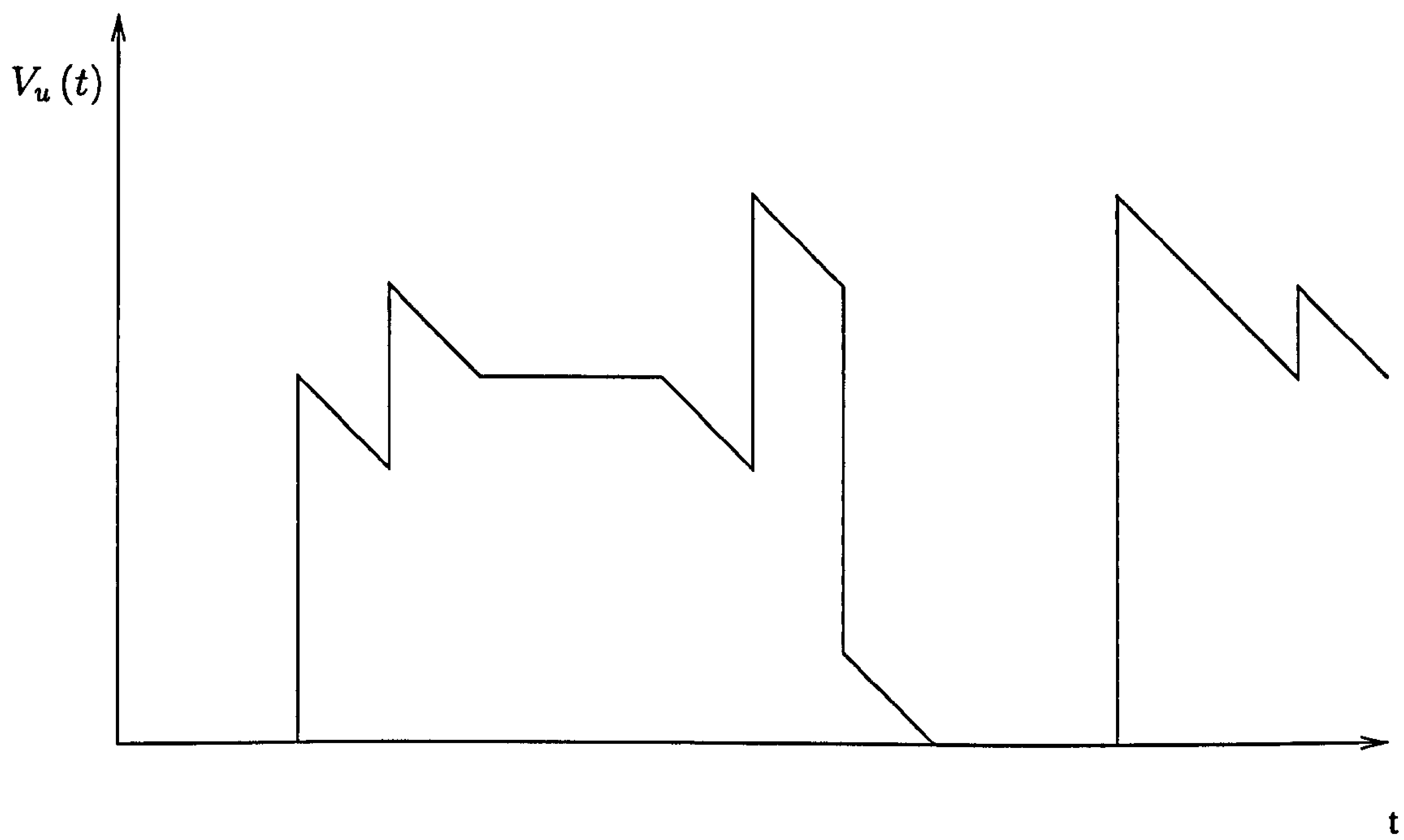

Figure 1.1: Total work in the system under scheduling strategy u.

vertically upwards by an amount equal to the time needed to service the incoming job: whenever a job is being served, it decreases linearly with slope -1. While the server is idle, $V_u(t)$ remains constant but jumps vertically downwards (by the amount of service time remaining) whenever a job leaves the system before it has completed its service. It is clear that the only influence which service strategy u can have on $V_u(t)$ is by ruling that the server should be idle when there are jobs in the system and by expelling jobs from the system before their service time is complete. (It is assumed that jobs do not leave the system of their own volition.) Under a strategy where the server is not allowed to idle and jobs not allowed to leave the system before service completion $V_u(t)$ is independent of u. Such strategies are termed *work-conserving*. Assume that $V_u(t)$, as a stochastic process, has a steady state distribution and denote the limiting average by V_u:

$$V_u = \lim_{t \to \infty} E\left[V_u(t)\right]. \tag{1.5}$$

As $V_u(t)$, and hence $E[V_u(t)]$, is independent of u for every t, it is, therefore, possible to state

$$V_u = V \tag{1.6}$$

where V is a constant, determined only by the parameters of the arrival and required service times processes. It is also assumed that all scheduling decisions are based on past and present knowledge of the system (i.e. not on any knowledge of exact remaining service times). Let $x_i^u = E_u(N_i)/\mu_i$ be the steady-state expected work in the system due to type i jobs and $\sum_{i=1}^{2} E_u(N_i)/\mu_i$ the total steady-state expected amount of work. In the 2 job-type system, therefore, we can conclude that

$$\sum_{i=1}^{2} \frac{E_u(N_i)}{\mu_i} = V = \frac{\rho_1 \mu_1^{-1} + \rho_2 \mu_2^{-1}}{1 - \rho_1 - \rho_2} \tag{1.7}$$

and it is also shown that

$$\sum_{i \in A} \frac{E_u(N_i)}{\mu_i} \geq \left[1 - \sum_{i \in A} \rho_i\right]^{-1} \sum_{i \in A} \frac{\rho_i}{\mu_i}, \quad \text{where } A = \{1\}, \{2\}, \tag{1.8}$$

for any admissible, work conserving, non-anticipative service policy where ρ_i, $i = 1, 2$ is the offered load for a type i job. The lower bounds for $A = \{1\}, \{2\}$ are obtained by policies which give pre-emptive priority to the appropriate class. This is an example of a system satisfying strong conservation laws. Bertsimas et al. (1994) sought to extend the range of problems by consideration of both open and closed multiclass queueing networks. Many of the problems they considered were such that it was not possible to characterise the associated achievable region exactly. Their interests led them to consider performance vectors which captured not only mean queue lengths, but also higher moments and higher order interactions between the customer classes. This will be of importance in the work of the thesis at various points. They used a potential function technique to account for higher order interactions among customer classes, or job types, and went on to obtain non-linear characterizations of relaxations of the achievable space using ideas from semi-definite programming. Ansell, Glazebrook, Mitrani & Niño-Mora (1999) adopted such an approach in their evaluation of performance policies for a two class queueing system. We have used such techniques throughout our work to evaluate the performances of a number of heuristic policies.

1.4.2 Generalised conservation laws.

The work of Shanthikumar and Yao was generalised by Bertsimas & Niño-Mora (1996). The latter pair showed that if the performance measures in stochastic and dynamic scheduling problems conform to certain *generalised conservation laws*, then the performance space is an *extended* polymatroid. Optimisation of a linear objective, over such a region, can be achieved by an adaptive greedy algorithm and yields an optimal solution with an *indexability* property. Generalised conservation laws extended the strong conservation laws of Definition (1.1) in that they introduce the concept of weighting to the performance vector. Again, we consider a stochastic service system where $E = \{1, \ldots, n\}$ be a set of n different job types . We assume scheduling strategies are non-idling and non-anticipative. Let x_j^u be a performance measure of type j jobs under an admissible policy u. We denote $\mathbf{x}^\pi$ the performance vector for the strict priority rule which prioritises the job types according to the permutation π of the n elements of set E. Thus, π_1 is given the highest priority, ... , π_n the lowest priority. We shall use the notation $x(S) := \sum_{j \in S} x(j)$, for any $S \subseteq E$.

Definition 1.3 (Generalised conservation laws) *Performance vector* $\mathbf{x}^u$ *is said to satisfy* generalised conservation laws *if* $\exists$ *a set function* b*:* $2^E \to \mathbb{R}^+$*, and a matrix*

$A = \left(A_j^S\right)_{j\in E, S\subseteq E}$ *that satisfies* $A_j^S > 0$, *for* $S \subseteq E$ *such that:*

(a)

$$b(S) = \sum_{j\in S} A_j^S x_j^\pi, \text{ for all } \pi : \{\pi_1, \dots, \pi_{|S^c|}\} = S^c \text{ and } S \subseteq E. \tag{1.9}$$

(b)

For every admissible policy $u \in U$ *;*

$$\sum_{j\in E} A_j^E x_j^u = b(E) \tag{1.10}$$

and

$$\sum_{j\in S} A_j^S x_j^u \geq b(S), \text{ all } S \subset E \tag{1.11}$$

As previously mentioned, the extension from strong conservation laws is by the addition of the weights A_j^S. Thus, when $A_j^S = 1$, *for all* $j \in S$ a performance vector will satisfy strong conservation laws. Consider the following polyhedron:

$$P(A, b) = \left\{ \mathrm{x} \in \left(\mathbb{R}^+\right)^n : \sum_{j\in S} A_j^S x_j \geq b(S) \text{ for } S \subset E \text{ and } \sum_{j\in E} A_j^E x_j = b(S) \right\}. \tag{1.12}$$

The performance space is (the base of) an extended polymatroid.

Theorem 1.2 (Performance region characterisation) *Assume the performance vector* x *satisfies generalised conservation laws (1.10) and(1.11) then*

- *The performance space is the (base of) an extended polymatroid,* $P(\mathrm{A}, b)$*, described by (1.12).*
- *The performance vectors of the absolute priority rules form the vertices of* $P(\mathrm{A}, b)$*, and* $x^\pi = v(\pi)$

Given an optimal scheduling problem,

$$Z^* = \inf_{u\in U} \sum_{j\in E} c_j x_j^u,$$

we can compute Z^* from the following linear programme

$$Z^* = \min_{x\in B(A,b)} \sum_{j\in E} c_j x_j^u. \tag{1.13}$$

Bertsimas & Niño-Mora (1996) use the following adaptive greedy algorithm, given in Table (1.1) as a solution method for the linear programming problem (1.13). The optimal

Table 1.1: The adaptive greedy algorithm.

INPUT: (c, A)

OUTPUT: $(\pi, \bar{\mathbf{y}}, \gamma)$, where

$\pi = (\pi_1, ..., \pi_E)$ is a ranking permutation of E

$\bar{\mathbf{y}} = (\bar{y}^S)_{S \subseteq E}$ is the optimal dual solution, and

$\gamma = (\gamma_1, ..., \gamma_E)$ are optimal allocation indices

STEP $k = E$

Set $S_1 = E$

Set $\bar{y}^{S_1} = min\left\{\frac{c_j}{A_j^E} : j \in S_1\right\}$

Pick $\pi_1 \in argmin\left\{\frac{c_j}{A_j^E} : j \in S_1\right\}$

Set $\gamma_{\pi_1} = \bar{y}^{S_1}$

STEP k. For $k = E - 1, ..., 1$

Set $S_k = S_{k-1} \setminus \{\pi_{k+1}\}$

Set

$$\bar{y}^{S_k} = min\left\{\frac{c_j - \sum_{i=1}^{k-1} A_j^{S_i} \bar{y}^{S_i}}{A_j^{S_k}} : j \in S_k\right\}$$

Pick

$$\pi_k \in argmin\left\{\frac{c_j - \sum_{i=1}^{k-1} A_j^{S_i} \bar{y}^{S_i}}{A_j^{S_k}} : j \in S_k\right\}$$

Set $\gamma_{\pi_k} = \gamma_{\pi_k} + \bar{y}^{S_k}$

STEP 0

For $S \subseteq E$:

Set $\bar{y}^S = 0$, if $S \notin \{S_1, ..., S_E\}$

ordering vector π corresponds to a Gittins index policy. Bertsimas & Niño-Mora (1996) show that the vector γ of the optimal priority indices is independent of how any ties are broken in the running of the algorithm. They further showed that $\mathbf{v}(\pi)$ and $\bar{\mathbf{y}}$ are an optimal primal-dual pair for the linear programme. Thus (1.13) is optimally solved by the performance measures achieved by adopting a service policy corresponding to π and scheduling problems which have linear objectives and satisfy generalised conservation laws are optimally solved by Gittins index policies. More recent work in this area has considered so-called *partial conservation laws* which extend the work to consider the indexability of more difficult problems such as restless bandits as outlined in Section 1.2.1. Niño-Mora (2001*b*) shows that if a set of partial conservation laws are satisfied, then the achievable region method may be used to optimally solve a stochastic scheduling problem, for a suitable range of linear performance objectives. The solution takes the form of a priority-index policy. He also investigates, using the same approach, the property of indexability of restless bandits as defined by Whittle (1988).

1.5 Non-linear holding costs

The commonly made assumption of linear holding costs has been called into question by Van Meighem (1995) and others. He argued that, in reality, non-linearity may arise from physical phenomena, such as the processing cost of perishable goods or from customer expectations. The latter arises in situations where the marginal cost to a firm of delaying a customer is greater if he is delayed beyond his expected delay time rather than within it.

Ansell, Glazebrook & Mitrani (2001) took a similar view. With pure priority policies in which total pre-emptive priority is given to one customer type, there is often the problem that the service offered to the lower priority traffic is unacceptably poor. Simulations have shown that service to the latter group tends to be poor on average and extremely variable.

In their work on threshold switching policies, Ansell et al. (2001) addressed this problem by imposing a constraint on the second moment of the queue length of the lower priority traffic in a standard stochastic optimisation problem. They then examined the relative performances of both randomised and threshold service policies using an M/M/1 model with two customer types.

In this thesis, we also consider the above problem and assess the performances of a number of policies based on a linear switching curve. We then extend this to a problem with a non-linear objective function, formulated as a restless bandit, and go on to find a Whittle index solution to the problem which we believe to be near optimal under certain conditions. More detail of this is given in the following section on thesis structure.

1.6 Thesis Outline

In this section we give an outline of the structure of the remaining chapters of the thesis. Throughout the thesis, we are concerned only with multi-class M/M/1 queueing systems. In such systems, it is generally known that, when seeking to optimise a linear cost function in the expected number of customers in each queue, strict priority policies are optimal. Such policies however, can have the disadvantage of large variation in queue lengths of lower priority customers. We seek to address such problems.

In Chapter 2, we are concerned only with two-class M/M/1 systems in which a linear holding cost objective has constraint(s) imposed on the second moments of queue length(s). We consider a class of service policy based on linear switching curves. Under such policies, the server continues to serve type 1 customers until the queue length of the type 2 customers reaches the line $n_2 = \alpha n_1 + \beta$ and is then switched to serving type 2. This develops the work of Ansell et al. (1999) on the class of threshold policies, a subgroup of policies based on a linear switching curve (i.e. those where $\alpha = 0$). Analytical techniques (the power series algorithm, conformal mapping and the epsilon algorithm) are outlined and then used to evaluate performance measures for policies based on switching curves and threshold policies. This enables us to search for the lowest costs achievable by the policy types. These costs are compared with a lower bound cost using the achievable region approach. We outline the methods used to characterise a set of achievable performance vectors for a relaxation of a performance region and go on to find such a set for our two customer system. This bounding set is then used in the formulation of a semidefinite program and solved using the SDPA (semidefinite programming algorithm) package developed by Fujisawa and Kojima. See Kojima (1994).

Despite the strong performance of the (optimal) policies based on linear switching curves, they are expensive in the amount of computational time employed in searching through α and β space to find the optimum.

Part of Chapter 3 and 4 has been published: see Ansell, Glazebrook, Niño-Mora & O'Keeffe (2003). In Chapter 3, we consider a multi-class M/M/1 system and associated optimisation problems. The system cost rate is additive across the customer classes and increasing convex in the numbers present within each class. A discounted version of the problem is formulated as a restless bandit problem. Such problems were introduced by Whittle (1988). He proposed an index-based heuristic for those problems meeting the requirement of indexability. Following Whittle, we develop an index for our multi-class queueing system. We show, by simple arguments, the form of the index for the discounted costs version of our queueing model and, by taking an appropriate limit, we then infer the appropriate index for the undiscounted problem of primary interest. In Chapter 4, we carry out a numerical investigation into the performance of index policies in cases involving quadratic costs for two and three customer systems. The analytical methods used are as in Chapter 2, but we also use the methods of dynamic programming via the value iteration algorithm (see Tijms (1994)) in order to calculate C^{OPT}, the minimum cost achievable by any admissible policy. We compare (i) the minimum cost achievable by any threshold policy with (ii) that achieved by any linear switching policy with (iii) that achieved by the Whittle index policy. These costs are in turn compared with (iv) C^{OPT} and (v) a semidefinite lower bound again calculated, as in Chapter 2, by utilising the achievable region approach.

Chapter 2

Application of policies based on linear switching curves to problems with non-linear constraints

2.1 Introduction

In queueing theory, much of the work on optimal service control in multi-class systems has aimed to determine policies which will minimise some measure of a system's overall cost rate. It has frequently been assumed that such cost rates are linear in the numbers of customers of each type present in the system. Thus, given an M/M/1 system with k customer classes and linear cost functions C_k, the marginal costs, C_k' are constants. It is probably fair to say, however, that often the prime motivation for making such assumptions lay in their rendering possible the analysis of otherwise intractable problems, rather than their representing a close approximation of reality. In his work on the generalised $c\mu$-rule, Van Meighem (1995) argues that it is the non-linear holding cost function which is to be found in many real-life systems and that the linear assumption is often simplistic. Non-linearity may be due to various causes. In the case of a factory producing goods, it may be due to the nature of the goods themselves. For example, the marginal cost of delaying the freezing of fresh vegetables will clearly increase at an increasing rate as time passes and the product begins to deteriorate. Another factor which may account for non-linearity in the holding costs is customer expectation. A certain level of delay, to varying degrees, is expected by all customers but once that expected level is surpassed, then there is the possibility, for example, that customers will in the long term withdraw their custom.

In multi-class queueing systems, such problems can be a feature of simple pre-emptive priority policies, where classes are given pre-emptive priority over other classes. Customers of lower priority classes are often compelled to suffer excessively long queues which are subject to extremely variable service. Ansell et al. (1999) seek to mitigate such problems by consideration of a stochastic optimisation model in the form of a two customer M/M/1 system which has constraints imposed on the second moments of the two queue lengths. Of the two families of parameterised heuristic policies which they analyse, the performance of the threshold policies was the more promising.

Ansell et al. (1999) investigated the performance of threshold policies and our work generalises theirs to include the whole family of policies determined by linear switching curves. It must be said that such an extension does involve the additional complexity of a second policy parameter. Our motivation also stems from Van Meighem (1995) in which he introduces a *generalised* $c\mu$-rule which, with non-decreasing convex delay costs, he shows is asymptotically optimal if the system operates in heavy traffic. Our work is in terms of holding costs, which depend on queue lengths rather than delay but, from Van Meighem (1995), we inferred that policies based on linear switching curves might work well for those systems where convex holding costs are quadratic functions of queue length. Such policies are explained in Section 2.4. The threshold policies considered by Ansell et al. (1999) are a special case of those policies based on a linear switching curve. In this chapter, motivated by these considerations, we extend the work of Ansell et al. (1999) by considering the performance of policies based on linear switching curves in several numerical examples, all of which concern a two customer type M/M/1 system as described in Section 2.3.

Our chief aims in this were as follows:

- To assess the performance of policies based on a linear switching curve in various holding cost minimisation problems, by measuring how closely the cost of operating such policies approaches a theoretical lower bound on the problem;
- To identify the type(s) of problem, if any, for which such policies are nearly optimal;
- To identify the type of problem in which the policies based on a linear switching curve significantly outperform threshold policies.

We would expect the two parameter policies to outperform the threshold policies, but clearly any reduction in costs offered by the former would have to be sufficiently large to account for the extra computing time that finding such a policy would take. Clearly, the time taken to find the lowest cost offered by a threshold policy, involving as it does a simple evaluation of the expected cost at each threshold value, is far shorter than any search over the parameters of intercept and slope which the general class of linear switching policies requires. The structure of the rest of this chapter is given below.

In Section 2.2, we explain generally, the methods we use in our analysis of multiclass queueing systems. These are the power series algorithm and the epsilon algorithm. In Section 2.4 we describe scheduling policies based on linear switching curves and go on to develop a set of balance equations to which we apply the methods of Section 2.2. This enables us to analyse a two customer type M/M/1 system as outlined in Section 2.3.

As our performance measures include both first and second moments of queue-length, we cannot characterise the exact achievable region. We therefore in Section 2.6 outline the methods of Bertsimas et al. (1994) in formulating a set of constraints to define a relaxation of the exact region and in Section 2.7 we formulate a bounding set for such a relaxation for the two customer system described in Section 2.3 and over which optimisation methods of semidefinite programming described in Section 2.8 can be undertaken to calculate the lower bound on achievable cost. The problems we consider involve the minimisation of a linear objective which we constrain with various forms of second moment constraints.

2.2 Analytical methods

The analysis of the stochastic processes underlying queueing systems is often restricted to that of the simplest processes. The problem of dimensionality means that it is often not practical to solve a set of balance equations for any but the simplest of systems. Thus, the use of efficient numerical methods in order to compute performance measures, such as expected queue lengths, is essential. We use a method first introduced by Hooghiemstra, Keane & van de Ree (1988) called the Power Series Algorithm (PSA) in which the balance equations are replaced by a set of equations which are recursively solvable. As we shall see, this involves the addition of one dimension to the state space. In the next section, we give an account of the general power-series algorithm based on the work of Blanc (1993).

2.2.1 The Power Series Algorithm

The PSA is a numerical method developed to compute performance measures of multi-queue type systems. It consists of power-series expansions of the state probabilities in terms of the load of a system. These expansions are used to recursively solve the global balance equations satisfied by these probabilities. Its precise application to the particular systems studied in this thesis are given as they arise in the text.

Consider that type of multi-class queueing system for which the stochastic queue length processes are multi-dimensional birth-death processes. We use the following notation: let R be the number of queues in the system and ρ be the traffic intensity or load on the system. It is the latter which is used as a variable in the power-series expansion.
Let $\rho a_j(\mathbf{n})$ and $d_j(\mathbf{n})$ be the respective arrival and departure rates to queue j when the system is in state $\mathbf{n}$, where $j = 1, 2, ..., R$. We use, n_j for the number of customers/jobs in queue j and $\mathbf{n} = (n_1, n_2, \ldots n_R)$. We denote by $p(\mathbf{n})$ the steady state probability that

the process, $\{\mathbf{N}(t), t \geq 0\}$, is in state $\mathbf{n} \in \mathbb{N}^R$, where $\mathbf{N}(t)$ represents the state of the system at time t.

We leave state $\mathbf{n}$ if:

- an arrival occurs at one of the queues or
- a service completion occurs at one of the queues.

Similarly, a state $\mathbf{n}$ is entered if:

- an arrival occurs at queue j when the system is in state $\mathbf{n} - \mathbf{e}_j$, for $n_j \geq 1$
- or a service completion occurs at queue j when the system is in state $\mathbf{n} + \mathbf{e}_j$, for $n_j \geq 1$.

We define $\mathbf{e}_j$ to be the unit vector, consisting of a component of one in the jth position, and all other components are zero $(j = 1, ..., R)$. Hence, the global balance equations for flows into and out of state $\mathbf{n}$ are as follows:

$$\left\{\rho \sum_{j=1}^{R} a_j(\mathbf{n}) + \sum_{j=1}^{R} d_j(\mathbf{n})\right\} p(\mathbf{n}) =$$
$$\rho \sum_{j=1}^{R} a_j(\mathbf{n} - \mathbf{e}_j)\, \delta\{n_j \geq 1\}\, p(\mathbf{n} - \mathbf{e}_j) + \sum_{j=1}^{R} d_j(\mathbf{n} + \mathbf{e}_j)\, p(\mathbf{n} + \mathbf{e}_j) \tag{2.1}$$

where $\delta(I)$ equals 1 if I holds and is 0 otherwise. We assume that the state probabilities sum to 1:

$$\sum_{n_1=0}^{\infty} \cdots \sum_{n_R=0}^{\infty} p(\mathbf{n}) = 1. \tag{2.2}$$

The state probabilities can be expanded as a power series in terms of ρ, the load of the system:

$$p(\mathbf{n}) = \rho^{|\mathbf{n}|} \sum_{k=0}^{\infty} \rho^k b(k; \mathbf{n}) \tag{2.3}$$

and substituting (2.3) into the global balance equations (2.1) results in the following recursive set of equations:

$$b(0;0) = 1;$$

$$\sum_{j=1}^{R} d_j(\mathbf{n})\, b(0,\mathbf{n}) = \sum_{j=1}^{R} a_j(\mathbf{n}-\mathbf{e}_j)\,\delta\{n_j \geq 1\}\, b(0,\mathbf{n}-\mathbf{e}_j),\ |\mathbf{n}| \geq 1 \tag{2.4}$$

where

$$|\mathbf{n}| = \sum_{i=1}^{R} n_i \text{ and } b(0;\mathbf{n}) = \lim_{\rho \downarrow 0} \rho^{-|\mathbf{n}|} p(\mathbf{n}).$$

The following relations must also be satisfied:

$$b(k;0) = -\sum_{1 \leq |\mathbf{n}| \leq k} b(k-|\mathbf{n}|;\mathbf{n});\ \ k = 1,2,... \tag{2.5}$$

and for $k = 1, 2, ...,\ \mathbf{n} \in \mathbb{N}^R,\ \mathbf{n} \neq 0$,

$$\begin{aligned} \sum_{j=1}^{R} d_j(\mathbf{n})\, b(k;\mathbf{n}) &= \sum_{j=1}^{R} a_j(\mathbf{n}-\mathbf{e}_j)\,\delta\{n_j \geq 1\}\, b(k;\mathbf{n}-\mathbf{e}_j) \\ &\quad - \sum_{j=1}^{R} a_j(\mathbf{n})\, b(k-1;\mathbf{n}) \\ &\quad + \sum_{j=1}^{R} d_j(\mathbf{n}+\mathbf{e}_j)\, b(k-1;\mathbf{n}+\mathbf{e}_j). \end{aligned} \tag{2.6}$$

The coefficients, $b(k;\mathbf{n})$, can be computed recursively from (2.4) - (2.6) if the stationary probabilities are rewritten in the form of (2.3). The power series obtained from these recursions does not always converge for all values of ρ for which the system is in steady state, i.e. $\rho < 1$. Blanc (1993) offers two methods of overcoming such problems by increasing the radius of convergence of the power series. They are briefly outlined in the following subsections.

2.2.2 Enlarging the radius of convergence

We can overcome problems of convergence by introducing a bilinear mapping of the interval [0, 1] on to itself. This has the effect of enlarging the radius of convergence of the power series. The conformal mapping used is as follows;

$$\theta = \Gamma_{G(\rho)} = \frac{(1+G)\rho}{1+G\rho}, \qquad \rho = \Gamma_G^{-1}(\theta) = \frac{\theta}{1+G-G\theta} \tag{2.7}$$

Choice of a suitable parameter, G, will allow the algorithm to converge for large values of ρ, see Blanc (1993) for details. An alternative computation scheme is then obtained by using the following power-series expansions in terms of θ instead of (2.3):

$$p(\mathbf{n}) = \theta^{|\mathbf{n}|} \sum_{k=0}^{\infty} \theta^k b_G(k;\mathbf{n}), \quad \mathbf{n} \in \mathbb{N}^R \tag{2.8}$$

By replacing ρ by $\frac{\theta}{1+G-G\theta}$ in the balance equations (2.1) and then substituting the expansions from (2.8) into these equations we produce the following set of recursive relations.

$$b_G(0;\mathbf{0}) = 1, \tag{2.9}$$

$$(1+G)\sum_{j=1}^{R} d_j(\mathbf{n})\, b_G(0;\mathbf{n}) = \sum_{j=1}^{R} a_j(\mathbf{n}-\mathbf{e}_j)\,\delta\{n_j \geq 1\}\, b_G(0;\mathbf{n}-\mathbf{e}_j), \quad |\mathbf{n}| \geq 1, \text{for } k=0,\ \mathbf{n} \in \mathbb{N}^R; \tag{2.10}$$

$$b_G(k;\mathbf{0}) = -\sum_{1\leq|\mathbf{n}|\leq k} b_G(k-|\mathbf{n}|;\ \mathbf{n}), \quad \text{for } k=1,2,...,\ \mathbf{n}=\mathbf{0}; \tag{2.11}$$

and

$$(1+G)\sum_{j=1}^{R} d_j(\mathbf{n})\, b_G(k;\mathbf{n}) =$$
$$\left\{\sum_{j=1}^{R} a_j(\mathbf{n}-\mathbf{e}_j)\,\delta\{n_j \geq 1\}\right\} - G\delta\{k \geq 2\}\, b_G(k-2;\mathbf{n}+\mathbf{e}_j))\}$$
$$+\sum_{j=1}^{R}\{G d_j(\mathbf{n}) - a_j(\mathbf{n})\}\, b_G(k-1;\mathbf{n})\, b_G(k;\mathbf{n}-\mathbf{e}_j)$$
$$+\sum_{j=1}^{R} d_j(\mathbf{n}+\mathbf{e}_j)\,\{(1+G)\, b_G(k-1;\mathbf{n}+\mathbf{e}_j)$$
$$\text{for } \mathbf{n} \in \mathbb{N}^R, \mathbf{n} \neq \mathbf{0}. \tag{2.12}$$

The choice of value for G depends on the radius of convergence of the given power series. A further method of improving convergence involves the use of the epsilon algorithm.

2.2.3 The epsilon algorithm

The epsilon algorithm accelerates the convergence of slowly convergent sequences, or can be used to calculate a value for divergent sequences. It does this by transforming an initial polynomial into quotients of two polynomials and consists of the following recursive scheme:

$$\epsilon_{\kappa}^{(m)} = \epsilon_{\kappa-2}^{(m+1)} + \left[\epsilon_{\kappa-1}^{(m+1)} - \epsilon_{\kappa-1}^{(m)}\right]^{-1}, \quad m \geq -\kappa, \ \ \kappa = 1, 2, ... \tag{2.13}$$

where the initial conditions are

$$\epsilon_{2\kappa}^{-\kappa-1} = 0, \ \ \kappa = 0, 1, ...; \quad \epsilon_{-1}^{(m)} = 0, \ \ \epsilon_{0}^{(m)} = \sum_{k=0}^{m} c_k \theta^k, m = 0, 1, ... \tag{2.14}$$

and where c_k, $k = 0, 1, 2, ...$ are the coefficients of a series. In our case, these coefficients are $b(k; \mathbf{n})$ and $b_G(k; \mathbf{n})$. It is only the even sequences which converge more rapidly than the initial sequence, the odd sequences being simply intermediate calculations. One problem which may arise in implementing the epsilon algorithm is that of computer storage capacity. It is often this rather than processing time which can limit the application of the PSA. Finding power-series expansions up to the Mth power of ρ (or θ as in (2.7)) requires the computation of

$$B_R(M) = \begin{pmatrix} M + R + 1 \\ R + 1 \end{pmatrix}$$

coefficients, $b(k; \mathbf{n})$; specifically, those with $k + |\mathbf{n}| \leq M$. Thus, in order to make efficient use of memory space, we map the multi-dimensional lattice points $(k; \mathbf{n})$, with $k + |\mathbf{n}| \leq M$, on to the integer set $\{0, ..., B_R(M) - 1\}$ using the one-to-one mapping given below:

$$C(k; \mathbf{n}) = \begin{pmatrix} k + |\mathbf{n}| + R \\ R + 1 \end{pmatrix} + \sum_{j=|\mathbf{n}|+1}^{|\mathbf{n}|+k} \begin{pmatrix} R + j - 1 \\ j \end{pmatrix} + \sum_{j=2}^{R} \begin{pmatrix} R - j + \sum_{i=j}^{R} n_i \\ R - j + 1 \end{pmatrix}. \tag{2.15}$$

This mapping has the necessary property that points $(k-1;\mathbf{n})$, $(k;\mathbf{n}-\mathbf{e}_j)$, $(k-1;\mathbf{n}+\mathbf{e}_j)$ and $(k-2;\mathbf{n}+\mathbf{e}_j)$, $j=1,...,R$, all map onto a value lower than that mapped onto by the point $(k;\mathbf{n})$ for $k=0,1,...,$ $\mathbf{n}\in\mathbb{N}^R$. Use of the algorithm enlarges the number of terms of the power-series expansions that can be computed with a given storage capacity. This is at the cost of the increased computation time which is needed to determine the location of the coefficients in the array in which they are stored.

2.3 Admissible Service Policies

Our aim is to carry out a performance analysis on two scheduling policies, one based on a threshold, the other based on a linear switching curve. A scheduling policy may be defined as a rule governing the allocation of the server(s) to customers in its queues. The scheduling policies which we consider are of a type defined as *admissible* in that they satisfy certain inherent restrictions.

A scheduling policy is deemed to be admissible if it is:

- Non-anticipative: there is no knowledge of the future of any aspect of the system so that decisions taken concerning server allocation can only be based on the history of the system to date;
- Work conserving: the server works whenever there are customers to be served and customers only leave the system when their processing has been completed

Let U denote the set of admissible service policies. Each service policy, $u\in U$ has associated with it a system performance vector, $\mathbf{x}_u=\{x_u^1,x_u^2,...,x_u^K\}$ where x_u^k is the expectation of some particular measure, such as queue length or waiting time, of the class k jobs.

We define the *performance space*, $\mathcal{X}=\{x_u : u\in U\}$, as the set of all system performances which can be achieved by the set of admissible policies. To this we add a cost rate vector, $\mathbf{c}=(c_1,c_2,...,c_K)^T$. Thus when we wish to optimise some aspect of the system's performance, we can express the problem as follows:

$$C^{opt} = \inf_{u \in U} \left(\mathbf{c}^T \mathbf{x}_u\right) = \inf_{\mathbf{x} \in X} \left(\mathbf{c}^T \mathbf{x}\right)$$

We seek to identify scheduling policies which correspond, subject to some form of constraints on the second moments of queue length, to the optimal solution of the above problems.

Throughout this chapter, we consider an M/M/1 queueing system with two customer types: type 1 and type 2. Arrivals occur in two independent Poisson streams with rates (λ_1, λ_2) for types 1 and 2 respectively. Service times are modelled via two exponential distributions with rates (μ_1, μ_2) again for types 1 and 2 respectively.

We assume that service policies must be non-anticipative, non-idling and pre-emptive. "Pre-emptive" means that if the server is busy serving a job of class k when a job of higher priority arrives, service can be switched instantaneously to the new arrival without any extra costs being incurred. This set up is summarised below in Figure 2.1.

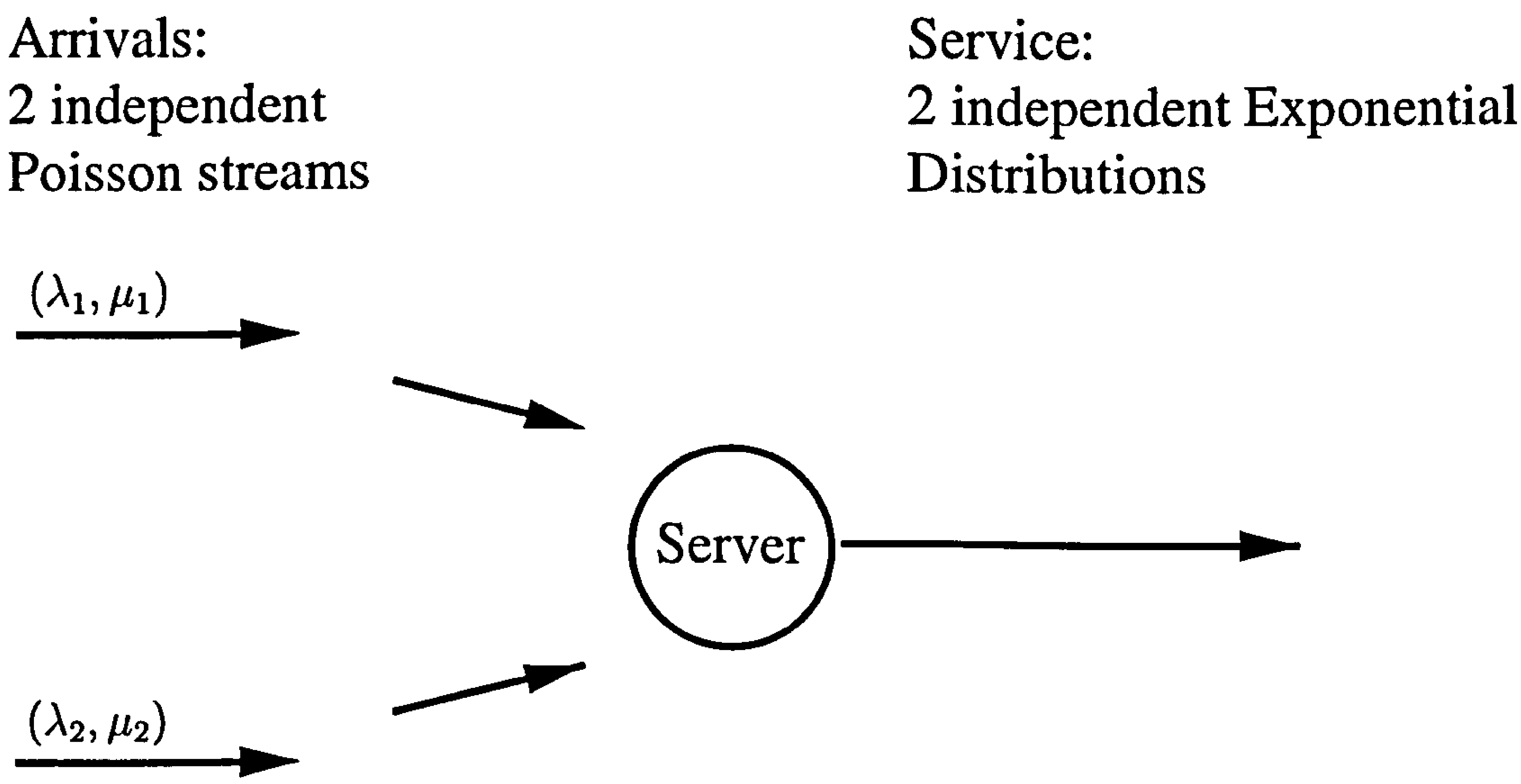

Figure 2.1: An M/M/1 system with two customer types.

We shall be concerned only with steady state performance criteria and therefore require the standard condition that $\rho_1 + \rho_2 = \frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2} < 1$. We measure performance by $[E_u(N_1), E_u(N_2), E_u(N_1^2), E_u(N_2^2)]$ where $E_u(N_k)$, and $E_u(N_k^2)$, $k = 1, 2$ are the expected queue lengths and second moments of queue lengths for type 1 and type 2 customers respectively, with expectation taken with respect to the stationary distribution under the chosen policy u.

From work done by Coffman & Mitrani (1980), the set of pairs $[E_u(N_1), E_u(N_2)]$ satisfy the following conditions for all policies u as illustrated in Figure 2.2 :

i $\frac{1}{\mu_1}E_u(N_1) + \frac{1}{\mu_2}E_u(N_2) = \left(\frac{\rho_1}{\mu_1} + \frac{\rho_2}{\mu_2}\right) / (1 - \rho_1 - \rho_2)$

ii $E_u(N_1) \geq \rho_1 / (1 - \rho_1)$

iii $E_u(N_2) \geq \rho_2 / (1 - \rho_2)$

and the set of achievable $[E_u(N_1), E_u(N_2)]$ is the line segment determined by:

$$H = \left[(x_1, x_2); x_1 \geq \frac{\rho_1}{(1-\rho_1)}, x_2 \geq \frac{\rho_2}{(1-\rho_2)}, \frac{x_1}{\mu_1} + \frac{x_2}{\mu_2} = \left(\frac{\rho_1}{\mu_1} + \frac{\rho_2}{\mu_2}\right) / (1 - \rho_1 - \rho_2)\right].$$

These are the Strong Conservation Laws for an M/M/1 system as described in Chapter 1.4.1 but these involve no second moments of queue lengths.

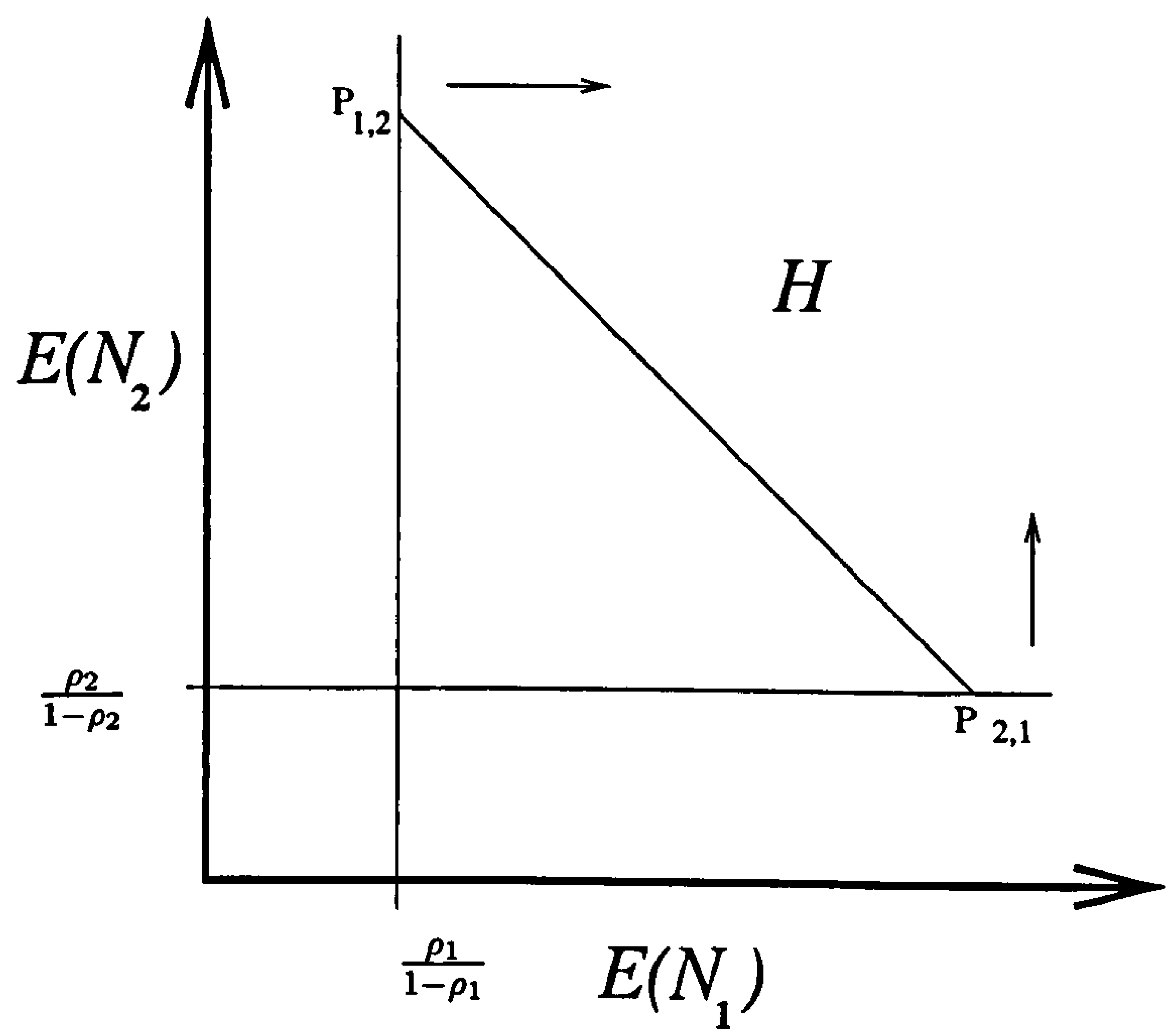

Figure 2.2: Achievable performance region, H

$P_{1,2}$ and $P_{2,1}$ correspond to the expected queue length pairs under the total priority policies for which pre-emptive priority is given to type 1 and type 2 customers respectively. However, the problems we consider include consideration of $E_u(N_1^2)$ and $E_u(N_2^2)$ and the exact achievable region is not available. More precisely, we seek to minimise

$$c_1 E_u(N_1) + c_2 E_u(N_2)$$

for some cost c_1, c_2 for those u meeting some given linear constraints on $E_u(N_1^2)$ and $E_u(N_2^2)$. We, therefore, in Section 2.6 construct bounding sets for such a relaxation of the achievable region developed from work by Bertsimas & Niño-Mora (1996).

2.4 Policies based on linear switching curves

Policies based on linear switching curves are characterised by two parameters, α and β, each ranging from $-\infty$ to ∞, which represent the slope and intercept respectively, of a line drawn on the positive quadrant. Priority is given to type 1 until the queue length of type 2 reaches the switching curve, the line $n_2 = \alpha n_1 + \beta$. If there are no type 1 jobs, service is given to type 2 and vice versa. This is shown in Figure 2.3. The threshold policies are those where $\alpha = 0$. Service effort is assigned as follows:

If $N_1(t) > 0$ and $N_2(t) > 0$ and $N_2(t) \leq \alpha N_1(t) + \beta$ then a type 1 customer is served.

If $N_1(t) > 0$ and $N_2(t) > 0$ and $N_2(t) > \alpha N_1(t) + \beta$ then a type 2 customer is served.

If $N_2(t) = 0$ and $N_1(t) > 0$ then a type 1 customer is served.

If $N_1(t) = 0$ and $N_2(t) > 0$ then a type 2 customer is served.

2.5 Performance analysis of policies based on linear switching curves using the power-series algorithm

We are again considering the classical single server queueing system with two customer types. Service allocation is pre-emptive. The performance measures which we seek to

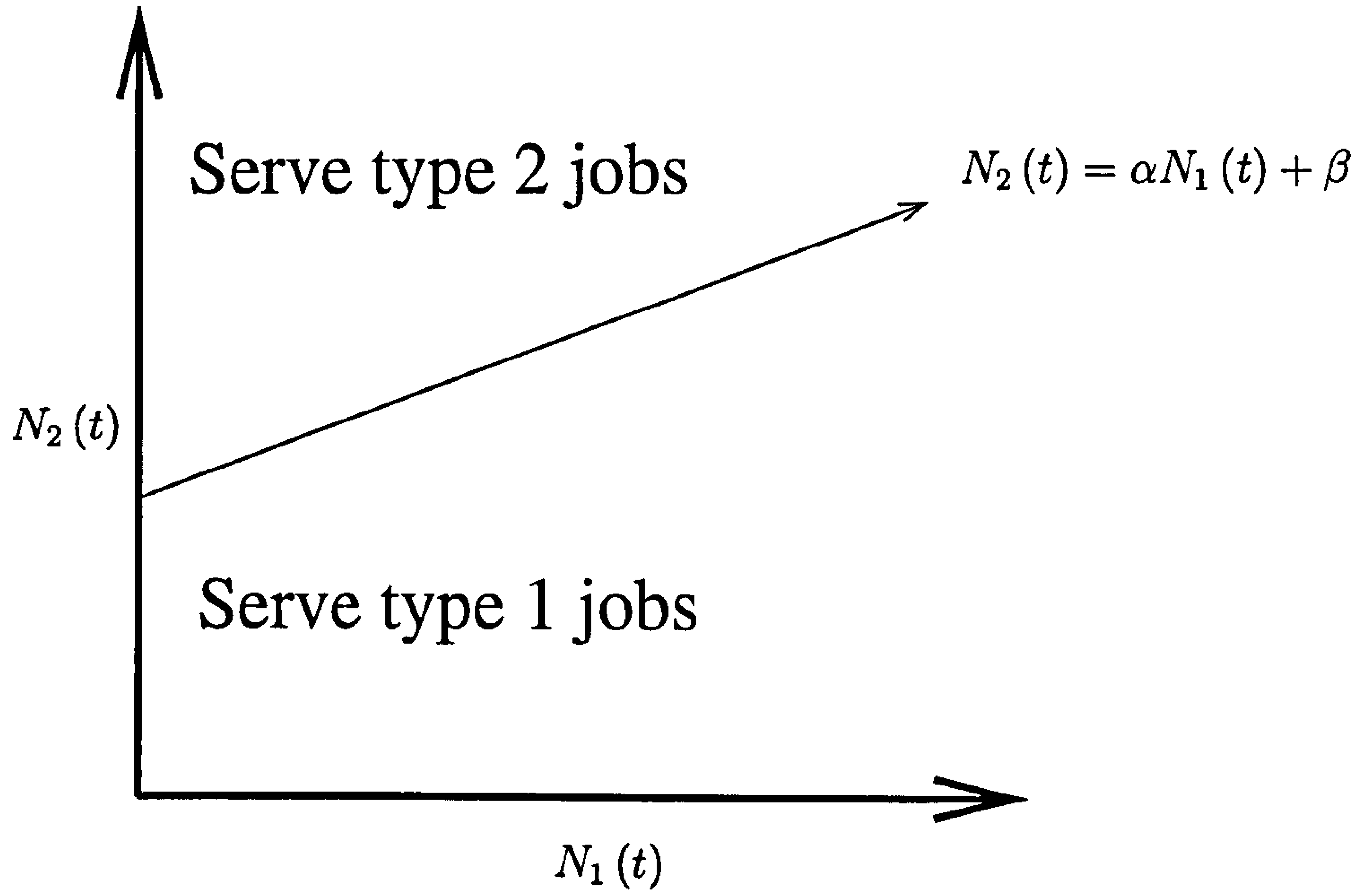

Figure 2.3: A Linear Switching Curve.

compute are the first and second moments of queue lengths under the linear switching curve policies in Section 2.4. For this we shall use the PSA as described in Section 2.2.1 for the case $R = 2$, where R is the number of customer types, in order to compute, $E_u(N_1)$,$E_u(N_1^2)$,$E_u(N_2)$ and $E_u(N_2^2)$ where u is a linear switching curve policy. The approach to the calculation of the first and second moments of the queue lengths of the system is described below. For the policies of interest, we need to determine the joint steady-state probability distribution

$$p_{i,j} = \lim_{t\to\infty} P\left[N_1(t) = i, N_2(t) = j\right] \tag{2.16}$$

and the above probabilities must satisfy the following set of balance equations:

$$\Big\{\lambda_1 + \lambda_2 + \mu_1\delta\,(i > 0, 0 < j \leq \alpha i + \beta) + \mu_2\delta\,(i > 0, j > 0, j > \alpha i + \beta)$$

$$+\mu_1\delta\,(i > 0, j = 0) + \mu_2\delta\,(i = 0, j > 0)\Big\}p_{i,j} =$$

$$\lambda_1 p_{i-1,j} + \lambda_2 p_{i,j-1} + \mu_1\delta\,(i+1 > 0, 0 < j \leq \alpha\,(i+1) + \beta)\,p_{i+1,j}$$

$$+\mu_1\delta\,(i+1 > 0, j = 0)\,p_{i+1,j} + \mu_2\delta\,(i = 0, j+1 > 0)\,p_{i,j+1}$$

$$+\mu_2\delta\,(i > 0, j+1 > 0, j+1 > \alpha i + \beta)\,p_{i,j+1}$$

where $p_{-1,j} = p_{i,-1} = 0$ and $\delta\,(B) = 1$ if B is true, 0 otherwise.

(2.17)

In our model, we have a two dimensional state space to describe the joint queue lengths for our two queue system. The parameter, $\rho = \rho_1 + \rho_2$ the load of the system is used as a variable in the power series expansion. We first rewrite the balance equations in the simpler form;

$$\Big\{\lambda_1 + \lambda_2 + \mu_1 L_1 + \mu_2 L_2 + \mu_1 L_3 + \mu_2 L_4\Big\}p_{i,j} \;=$$
$$\lambda_1 p_{i-1,j} + \lambda_2 p_{i,j-1} + \mu_1\,(L_5 + L_6)\,p_{i+1,j} + \mu_2\,(L_7 + L_8)\,p_{i,j+1} \qquad (2.18)$$

where $L_1 = \delta\,(i > 0, 0 < j \leq \alpha i + \beta)$;

$L_2 = \delta\,(i > 0, j > 0, j > \alpha i + \beta)$;

$L_3 = \delta\,(i > 0, j = 0)$;

$L_4 = \delta\,(i = 0, j > 0)$;

$L_5 = \delta\,(i+1 > 0, 0 < j \leq \alpha\,(i+1) + \beta)$;

$L_6 = \delta\,(i+1 > 0, j = 0)$;

$L_7 = \delta\,(i > 0, j+1 > 0, j+1 > \alpha i + \beta)$;

$L_8 = \delta\,(i = 0, j+1 > 0)$.

The conformal mapping for the balance equations is as follows:

$$\begin{aligned}\left\{ (\lambda_1+\lambda_2)\frac{\theta}{1+G-G\theta} + \mu_1 L_1 + \mu_2 L_2 + \mu_1 L_3 + \mu_2 L_4 \right\} p_{i,j} &= \\ &\left\{ \lambda_1 p_{i-1,j} + \lambda_2 p_{i,j-1} \right\} \frac{\theta}{1+G-G\theta} \\ &+\mu_1 (L_5+L_6) p_{i+1,j} + \mu_2 (L_7+L_8) p_{i,j+1}\end{aligned}$$

Using the conformed mapping approach, we now replace the $p_{i,j}$ so that we achieve the power series form. We use

$$p_{i,j} = \theta^{i+j} \sum_{h=0}^{\infty} \theta^h p_{h,i,j}$$

and we infer that

$$\begin{aligned}&\left\{ (\lambda_1+\lambda_2)\frac{\theta}{1+G-G\theta} + \mu_1 L_1 + \mu_2 L_2 + \mu_1 L_3 + \mu_2 L_4 \right\} \theta^{i+j} \sum_{h=0}^{\infty} \theta^h p_{h,i,j} \\ &= \left\{ \lambda_1 \theta^{i-1+j} \sum_{h=0}^{\infty} \theta^h p_{h,i-1,j} + \lambda_2 \theta^{i+j-1} \sum_{h=0}^{\infty} \theta^h p_{h,i,j-1} \right\} \frac{\theta}{1+G-G\theta} \\ &+\mu_1 (L_5+L_6)\, \theta^{i+1+j} \sum_{h=0}^{\infty} \theta^h p_{h,i+1,j} \\ &+\mu_2 (L_7+L_8)\, \theta^{i+j+1} \sum_{h=0}^{\infty} \theta^h p_{h,i,j+1}.\end{aligned} \tag{2.19}$$

Now, we multiply through by $(1+G-G\theta)$ to obtain

$$\begin{aligned}\left\{ (\lambda_1+\lambda_2)\,\theta^{i+j+1} + \{\mu_1 (L_1+L_3) + \mu_2 (L_2+L_4)\}\,(1+G-G\theta)\,\theta^{i+j} \right\} \sum_{h=0}^{\infty} \theta^h p_{h,i,j} &= \\ &\lambda_1 \theta^{i+j} \sum_{h=0}^{\infty} \theta^h p_{h,i-1,j} + \lambda_2 \theta^{i+j} \sum_{h=0}^{\infty} \theta^h p_{h,i,j-1} \\ &+\mu_1 (L_5+L_6)\, \theta^{i+1+j} (1+G-G\theta) \sum_{h=0}^{\infty} \theta^h p_{h,i+1,j} \\ &+\mu_2 (L_7+L_8)\, \theta^{i+j+1} (1+G-G\theta) \sum_{h=0}^{\infty} \theta^h p_{h,i,j+1}.\end{aligned}$$

Taking out a factor of θ^{i+j} gives

$$\left\{ (\lambda_1+\lambda_2)\,\theta + \left\{\mu_1\,(L_1+L_3)+\mu_2\,(L_2+L_4)\right\}(1+G-G\theta)\right\}\sum_{h=0}^{\infty}\theta^h p_{h,i,j}$$

$$\begin{aligned} &= \lambda_1\sum_{h=0}^{\infty}\theta^h p_{h,i-1,j}+\lambda_2\sum_{h=0}^{\infty}\theta^h p_{h,i,j-1} \\ &\quad +\mu_1\,(L_5+L_6)\,\theta\,(1+G-G\theta)\sum_{h=0}^{\infty}\theta^h p_{h,i+1,j} \\ &\quad +\mu_2\,(L_7+L_8)\,\theta\,(1+G-G\theta)\sum_{h=0}^{\infty}\theta^h p_{h,i,j+1}. \end{aligned}$$

Rearranging, this gives:

$$\begin{aligned} &\Big\{ (\lambda_1+\lambda_2)\,\theta+\mu_1\,(L_1+L_3)\,(1+G)-\mu_1\,(L_1+L_3)\,G\theta \\ &+\mu_2\,(L_2+L_4)\,(1+G)-\mu_2\,(L_2+L_4)\,G\theta\Big\}\sum_{h=0}^{\infty}\theta^h p_{h,i,j} = \\ &\quad \lambda_1\sum_{h=0}^{\infty}\theta^h p_{h,i-1,j}+\lambda_2\sum_{h=0}^{\infty}\theta^h p_{h,i,j-1} \\ &\quad +\mu_1\,(L_5+L_6)\,\theta\,(1+G)\sum_{h=0}^{\infty}\theta^h p_{h,i+1,j}-\mu_1\,(L_5+L_6)\,G\theta^2\sum_{h=0}^{\infty}\theta^h p_{h,i+1,j} \\ &\quad +\mu_2\,(L_7+L_8)\,\theta\,(1+G)\sum_{h=0}^{\infty}\theta^h p_{h,i,j+1}-\mu_2\,(L_7+L_8)\,G\theta^2\sum_{h=0}^{\infty}\theta^h p_{h,i,j+1}. \end{aligned}$$

Equate those terms with equal powers of θ to obtain:

$$\begin{aligned} &(\lambda_1+\lambda_2)\,p_{h-1,i,j}+\mu_1\,(L_1+L_3)\,(1+G)\,p_{h,i,j}-\mu_1\,(L_1+L_3)\,Gp_{h-1,i,j} \\ &+\mu_2\,(L_2+L_4)\,(1+G)\,p_{h,i,j}-\mu_2\,(L_2+L_4)\,Gp_{h-1,i,j} = \\ &\qquad \lambda_1 p_{h,i-1,j}+\lambda_2 p_{h,i,j-1} \\ &\qquad +\mu_1\,(L_5+L_6)\,(1+G)\,p_{h-1,i+1,j}-\mu_1\,(L_5+L_6)\,Gp_{h-2,i+1,j} \\ &\qquad +\mu_2\,(L_7+L_8)\,(1+G)\,p_{h-1,i,j+1}-\mu_2\,(L_7+L_8)\,Gp_{h-2,i,j+1}. \end{aligned}$$

Thus we have

$$\begin{aligned} p_{h,i,j} &= \Big\{ \lambda_1 p_{h,i-1,j} + \lambda_2 p_{h,i,j-1} - (\lambda_1 + \lambda_2) p_{h-1,i,j} \\ &+ \mu_1 (L_5 + L_6)(1+G) p_{h-1,i+1,j} - \mu_1 (L_5 + L_6) G p_{h-2,i+1,j} \\ &+ \mu_2 (L_7 + L_8)(1+G) p_{h-1,i,j+1} - \mu_2 (L_7 + L_8) G p_{h-2,i,j+1} \\ &+ \mu_1 (L_1 + L_3) G p_{h-1,i,j} + \mu_2 (L_2 + L_4) G p_{h-1,i,j} \Big\} \\ & \left\{ \left[\mu_1 (L_1 + L_3) + \mu_2 (L_2 + L_4) \right] (1+G) \right\}^{-1}. \end{aligned}$$

We then use the epsilon algorithm to improve convergence. As we have stated earlier, our aim is to compare the performances of the various heuristic policies and also to assess their performance against the optimum achievable cost. It is not possible to characterise the exact achievable region for the first and second moments of queue length. We therefore must formulate a set of constraints which will yield a relaxation of this region.

2.6 Bounding sets

The goal of the analysis is to develop sets of equations/inequalities which are satisfied by the first and second moments of queue length under all policies. These can then be used to develop a relaxation of the required achievable region. We use the non-parametric bounding method put forward by Bertsimas et al. (1994). They consider a network consisting of T stations, populated by R classes of job. The class of a job completely summarizes all of its characteristics, including the node (server) at which it is awaiting service. Jobs awaiting service at different nodes are by definition of different classes and it follows thereby that a job changes class whenever it moves from one node to another in the network. We use $\sigma(r)$ to represent the node at which class r customers are served and C_i is the set of all classes, r, such that $\sigma(r) = i$. When a class r job completes service at node i, it becomes a job of class s with probability p_{rs} and so moves to server $\sigma(s)$ or it can exit the network with probability $p_{ro} = 1 - \sum_{s=1}^{R} p_{rs}$. Policies considered are non-anticipative, pre-emptive but not necessarily work-conserving. The number of class r customers in the system at time t is denoted by $n_r(t)$ and $\mathbf{n}(t) = (n_1(t), n_2(t), ..., n_R(t))$ represents the state of the system at time t. Under a Markovian policy, such a queueing network will evolve as a continuous-time Markov chain. We assume that $\mathbf{n}(t)$ has a unique steady-state distribution with mean vector $\mathbf{n} = \{n_1, n_2, ..., n_R\}$. We also assume that $E[n_r^2(t)] < \infty$, i.e. the expectation of the second moments are finite when taken with respect to the steady-state distribution. The goal is to determine the *region of achievable performance*, i.e. the set of all mean vectors $\mathbf{n} = (n_1, n_2, ..., n_R)$ obtained under different policies and in our work we also wish to include second moment vectors. The exact characterisation of the achievable region is not possible in general (see Bertsimas et al. (1994)) and so they devised methods which approximate the region by a larger set. It is by then minimising over this relaxation of the performance space that a lower bound may be found on the cost of an optimal policy. This is the approach we use. We define a potential function of the form:

$$R(t) = \sum_{r=1}^{R} f(r) n_r(t). \tag{2.20}$$

The following notation is used: τ_k is the sequence of transition times in a uniformised Markov chain such that $\sum_r \lambda_{or} + \sum_r \mu_r = 1$. $B_r(t)$ denotes the event that server $\sigma(r)$ is busy with a class r customer at time t. Similarly, $\overline{B}_r(t)$ denotes the event that $\sigma(r)$ is *not* busy with a class r customer at time t. $\overline{B}_{0i}(t)$ denotes the event that node i is idle at time t. The arrival process for class r customers has rate λ_{0r} and the service time of class r jobs is assumed to have an exponential distribution with rate μ_r. We define

$$I_{rr'} = E[\delta(B_r\{\tau_k\}) n_{r'}(\tau_k)], \tag{2.21}$$

$$N_{ir'} = E[\delta(B_{0i}\{\tau_k\}) n_{r'}(\tau_k)], \tag{2.22}$$

where $\delta(.)$ is the indicator function and the expectations are taken with respect to the invariant distribution. We obtain a recursion as follows:

$$E\left[R^2(\tau_{k+1})\,|\mathbf{n}(\tau_k)\right] = \sum_{r=1}^{R} \lambda_{0r}\left(R(\tau_k) + f(r)\right)^2 + \sum_{r=1}^{R} \mu_r \delta\left(\overline{B}_r\{\tau_k\}\right) R^2(\tau_k)$$
$$+\sum_{r=1}^{R} \mu_r \delta\left(B_r\{\tau_k\}\right)\left[\sum_{r'=1}^{R} p_{rr'}\left(R(\tau_k) - f(r) + f(r')\right)^2 + p_{r0}\left(R(\tau_k) - f(r)\right)^2\right] \tag{2.23}$$

$$E\left[R(\tau_k)\right] = E\left[\sum_{r=1}^{R} n_r(\tau_k) f(r)\right] = \sum_{r=1}^{R} f(r)\, n_r \tag{2.24}$$

$$E[\delta(B_r\{\tau_k\})] = \frac{\lambda_r}{\mu_r},. \tag{2.25}$$

From (2.20)-(2.25), we can see that

$$E\left[\delta(B_r\{\tau_k\})\,R(\tau_k)\right] = \sum_{r'=1}^{R} f(r')\, I_{rr'}. \tag{2.26}$$

Now substitute (2.20)-(2.25) into (2.23), take expectations and equate those terms in $\{f(r)\}^2$ and $\{f(r)\,f(r')\}$ we have the following:

Theorem 2.1 *The following equalities hold true for every scheduling policy which satisfies the above assumptions:*

$$2\mu_r I_{rr} - 2\sum_{r'=1}^{R} \mu_{r'} p_{r'r} I_{r'r} - 2\lambda_{0r} n_r = \lambda_{0r} + \lambda_r(1 - p_{rr}) + \sum_{r' \neq r} \lambda_{r'} p_{r'r}, \quad 1 \le r \le R$$

$$\mu_r I_{rr'} + \mu_{r'} I_{r'r} - \sum_{w=1}^{R} \mu_w p_{wr} I_{wr'} - \sum_{w=1}^{R} \mu_w p_{wr'} I_{wr} - \lambda_{0r} n_{r'} - \lambda_{0r'} n_r = -\lambda_r p_{rr'} - \lambda_{r'} p_{r'r} \quad \textit{for all } r, r' \textit{ such that } r > r'.$$

$$\sum_{r \in C_i} I_{rr'} + N_{ir'} = n_{r'}, \qquad I_{rr'} \ge 0, \quad N_{ir'} \ge 0, \quad n_i \ge 0.$$

2.6.1 Higher order interactions

The methodology can be extended to take account of higher moments and higher order interactions, as opposed to the pairwise interactions expressed in (2.21) and (2.22) and derived so far. In the case of pairwise interactions, recursions were developed from a recursion for $E\left[R^2(\tau_{k+1})\,|\mathbf{n}(\tau_k)\right]$. Now, for higher order interactions, they are obtained, in a similar fashion from a recursion for $E\left[R^3(\tau_{k+1})\,|\mathbf{n}(\tau_k)\right]$.

It is necessary to introduce some new variables, namely,:

$$M_{st} = E\left[n_s(\tau_k)\, n_t(\tau_k)\right], \tag{2.27}$$

$$H_{rst} = E\left[\delta\left(B_r\{\tau_k\}\right) n_s(\tau_k)\, n_t(\tau_k)\right]. \tag{2.28}$$

We obtain the recursion:

$$\begin{aligned} E\left[R^3(\tau_{k+1})\,|\mathbf{n}(\tau_k)\right] &= \sum_{r=1}^{R} \lambda_{0r}\left(R(\tau_k) + f(r)\right)^3 + \mu_r \sum_{r=1}^{R} \delta\left(\overline{B}_r\{\tau_k\}\right) R^3(\tau_k) \\ &+ \sum_{r=1}^{R} \mu_r \delta\left(B_r\{\tau_k\}\right) \left\{ \sum_{r'=1}^{R} p_{rr'}\left(R(\tau_k) + f(r') - f(r)\right)^3 + p_{r0}\left(R(\tau_k) - f(r)\right)^3 \right\}. \end{aligned}$$

Expectations can now be taken with respect to the system in steady state. We have that

$$\begin{aligned} 0 &= 3\sum_{r=1}^{R} \lambda_{0r} f(r)\, E\left[R^2(\tau_k)\right] + 3\sum_{r=1}^{R} \lambda_{0r} f^2(r)\, E\left[R(\tau_k)\right] + \sum_{r=1}^{R} \lambda_{0r} f^3(r) \\ &+ 3\sum_{r=1}^{R} \mu_r \left[\sum_{r'=1}^{R} p_{rr'}\left(f(r') - f(r)\right)\right] E\left[\delta\left(B_r\{\tau_k\}\right) R^2(\tau_k)\right] \\ &+ 3\sum_{r=1}^{R} \mu_r \left[\sum_{r'=1}^{R} p_{rr'}\left(f^2(r') - 2f(r') f(r) + f^2(r)\right)\right] E\left[\delta\left(B_r\{\tau_k\}\right) R(\tau_k)\right] \\ &+ \sum_{r=1}^{R} \mu_r \left[\sum_{r'=1}^{R} p_{rr'}\left(f^3(r') - 3f^2(r') f(r) + 3f(r') f^2(r) - f^3(r)\right)\right] E\left[\delta\left(B_r\{\tau_k\}\right)\right] \\ &+ 3\sum_{r=1}^{R} \mu_r f(r)\, p_{r0} E\left[\delta\left(B_r\{\tau_k\}\right) R^2(\tau_k)\right] \\ &+ 3\sum_{r=1}^{R} \mu_r f^2(r)\, p_{r0} E\left[\delta\left(B_r\{\tau_k\}\right) R(\tau_k)\right] - \sum_{r=1}^{R} \mu_r f^3(r)\, p_{r0} E\left[\delta\left(B_r\{\tau_k\}\right)\right]. \end{aligned} \tag{2.29}$$

It is trivial to show that the following identities are satisfied

$$E[R^2(\tau_k)] = E\left[\sum_{r=1}^{R} n_r(\tau_k)\, f(r)\right]^2 = \sum_{r=1}^{R} f^2(r)\, M_{rr} + 2\sum_{r=1}^{R}\sum_{s=r+1}^{R} f(r) f(s)\, M_{rs}, \tag{2.30}$$

$$E\left[\delta\left(B_r\{\tau_k\}\right)\left[\sum_{s=1}^{R} f(s)\, n_s(\tau_k)\right]^2\right] = \sum_{s=1}^{R} f^2(s)\, H_{rss} + 2\sum_{r=1}^{R}\sum_{t=s+1}^{R} f(s) f(t)\, H_{rst}, \tag{2.31}$$

Substituting (2.24), (2.25), (2.30) and (2.31) into (2.29), we obtain:

$$\begin{aligned}
0 &= 3\sum_{r=1}^{R}\lambda_{0r}f(r)\left[\sum_{s=1}^{R}f^{2}(s)M_{ss}+2\sum_{s=1}^{R}\sum_{t=s+1}^{R}f(s)f(t)M_{st}\right] \\
&\quad +3\sum_{r=1}^{R}\lambda_{0r}f^{2}(r)\sum_{s=1}^{R}f(s)n_{s}+\sum_{r=1}^{R}\lambda_{0r}f^{3}(r) \\
&\quad +3\sum_{r=1}^{R}\mu_{r}\left[\sum_{s=1}^{R}p_{rs}\left(f^{2}(s)-2f(s)f(r)+f^{2}(r)\right)\right]\left[\sum_{t=1}^{R}f(t)I_{rt}\right] \\
&\quad +3\sum_{r=1}^{R}\mu_{r}\left[\sum_{s=1}^{R}p_{rs}\left(f(s)-f(r)\right)\right]\times\left[\sum_{t=1}^{R}f^{2}(t)H_{rtt}+2\sum_{t=1}^{R}\sum_{w=s+1}^{R}f(t)f(w)H_{rtw}\right] \\
&\quad +\sum_{r=1}^{R}\lambda_{r}\left[\sum_{s=1}^{R}p_{rs}\left(f^{3}(s)-3f^{2}(s)f(r)+3f(s)f^{2}(r)-f^{3}(r)\right)\right] \\
&\quad -3\sum_{r=1}^{R}\mu_{r}f(r)p_{r0}\left[\sum_{s=1}^{R}H_{rss}f^{2}(s)+2\sum_{s=1}^{R}\sum_{t=s+1}^{R}f(s)f(t)H_{rst}\right] \\
&\quad +3\sum_{r=1}^{R}\mu_{r}f^{2}(r)p_{r0}\left[\sum_{s=1}^{R}f(s)I_{rs}\right]-\sum_{r=1}^{R}\lambda_{r}f^{3}(r)p_{r0}. \qquad (2.32)
\end{aligned}$$

The r.h.s. of (2.29) is identically equal to 0 for all of the f-parameters. Therefore, we equate coefficients of powers of f to zero to obtain sets of equations. First, we equate the R coefficients of the terms $f^{3}(i)$ to obtain

$$\begin{aligned}
0 &= 3\lambda_{0i}M_{ii}+3\lambda_{0i}n_{i}+\lambda_{0i}+3\mu_{i}\sum_{s\neq i}^{R}p_{is}I_{is}+3\mu_{i}p_{i0}I_{ii}+3\sum_{s\neq i}^{R}\mu_{s}p_{si}I_{si} \\
&\quad +3\sum_{r'\neq i}^{R}\mu_{r'}p_{r'i}H_{r'ii}-3\mu_{i}p_{i0}H_{iii}-3\mu_{i}\sum_{r'\neq i}^{R}p_{ir'}H_{r'ii} \\
&\quad +3\mu_{i}p_{ii}H_{iii}-3\mu_{i}p_{ii}H_{iii}-\lambda_{i}+\sum_{r'=1}^{R}\lambda_{r'}p_{r'i} \qquad i=1,\ldots,R. \qquad (2.33)
\end{aligned}$$

Noting that by normalisation

$$\mu_{i}\sum_{s=1}^{R}p_{is}I_{is}=1 \qquad (2.34)$$

and

$$\mu_{i}\sum_{s=1}^{R}p_{is}H_{sii}=1, \qquad (2.35)$$

we can rewrite (2.33) as:

$$\begin{aligned} 0 &= -\lambda_i + \sum_{s=1}^{R} \lambda_s p_{si} + 3\mu_i (1 - p_{ii}) I_{ii} + 3 \sum_{s \neq i}^{R} \mu_s p_{si} I_{si} - 3\mu_i H_{iii} \\ &\quad + 3 \sum_{s=1}^{R} \mu_s p_{si} H_{sii} + \lambda_{0i} + 3\lambda_{0i} n_i + 3\lambda_{0i} M_{ii} \qquad i = 1, ..., R. \end{aligned} \tag{2.36}$$

We now equate the $R(R-1)$ coefficients of $f^2(i) f(j)$, for $i \neq j$ and by using the identities

$$\sum_{s=1}^{R} p_{is} I_{is} = 1 \tag{2.37}$$

and

$$\sum_{s=1}^{R} p_{is} H_{iis} = 1, \tag{2.38}$$

we produce the following set of constraints:

$$\begin{aligned} 0 &= 3\lambda_{0j} M_{ii} + 3\lambda_{0i} n_j + 6\lambda_{0i} M_{ij} + 3\lambda_i p_{ij} - 3\lambda_j p_{ji} + 3\mu_i (1 - p_{ii}) I_{ij} \\ &\quad - 6\mu_i p_{ij} I_{ii} - 6\mu_j p_{ji} I_{ji} + 3 \sum_{w \neq i}^{R} \mu_w p_{wi} I_{wj} + 3 \sum_{l=i}^{R} \mu_l p_{lj} H_{lii} \\ &\quad - 3\mu_j H_{jii} - 6\mu_i H_{iij} + 6 \sum_{w=1}^{R} \mu_w p_{wi} H_{wij}, \qquad \textit{all } i, j \textit{ s.t. } i \neq j. \end{aligned} \tag{2.39}$$

The final set of constraints (2.40) are derived from equating the $\sum_{w=1}^{R} (w-1)(w-2)/2$ coefficients of terms with $f(i) f(j) f(k), i \neq j, j \neq k$ and $i \neq k$, namely

$$\begin{aligned} 0 &= \lambda_{0k} M_{ij} + \lambda_{0i} M_{jk} + \lambda_{0j} M_{ik} - \mu_i (1 - p_{ii}) H_{ijk} - \mu_j (1 - p_{jj}) H_{jik} - \mu_k (1 - p_{kk}) H_{kij} \\ &+ \sum_{l \neq i,j,k}^{R} \mu_l p_{li} H_{ljk} + \sum_{l \neq i,j,k}^{R} \mu_l p_{lj} H_{lik} + \sum_{l \neq i,j,k}^{R} \mu_l p_{lk} H_{lij} - \mu_i p_{ij} I_{ik} + \mu_i p_{ij} H_{iik} \\ &- \mu_i p_{ik} I_{ij} + \mu_i p_{ik} H_{iij} - \mu_j p_{jk} I_{jk} + \mu_j p_{ji} H_{jjk} - \mu_j p_{jk} I_{ji} + \mu_j p_{jk} H_{jji} \\ &- \mu_k p_{ki} I_{kj} + \mu_k p_{ki} H_{kkj} - \mu_k p_{kj} I_{ki} + \mu_k p_{kj} H_{kki} \quad \textit{all } i, j, k \textit{ s.t. } i \neq j,\ i \neq k \text{ and } j \neq k. \end{aligned} \tag{2.40}$$

The following constraints also apply:

$$\sum_{l \in C_m} I_{li} \leq n_i, \qquad i = 1, ..., R, m = 1, ..., T, \tag{2.41}$$

$$\sum_{l \in C_m} H_{ljk} \leq M_{jk}, \qquad j,k = 1,...,R, m = 1,...,T, \tag{2.42}$$

$$n_i, I_{ij}, M_{jk}, H_{ijk} \geq 0. \tag{2.43}$$

Thus, a new set of linear constraints has been developed. These constraints involve $\{n_i, I_{ij}, M_{jk}, H_{ijk}\}$ and will allow us to develop a relaxation of the achievable region to consider problems involving second moment constraints. P_2 is the set of equations given in Theorem 2.1 and P_3 is the set defined by (2.40)-(2.43). It would be possible to apply the non-parametric method to $E\left[R^i\left(\tau_{k+1}\right)\right]$ for $i \geq 4$. In such a way it is possible to model interactions among i classes in the system. Such increases in accuracy, however, come with the 'cost' of reduced tractability. If the relaxation obtained by considering a recursion for the expectation of the ith power of the potential function is denoted by P_i, then the ith order approximation of the achievable performance region is said to be $\bigcap_{l=2}^{i} P_l$. By solving the related problem over a third order relaxation of the achievable region, $\bigcap_{l=2}^{3} P_l$ derived by the potential function method, we can find a lower bound on the optimal cost for the original scheduling problem.

2.7 A relaxation of the achievable region in a two customer class system

We now seek to assess the performance of various policies/systems as described in Section (2.3) for our cost minimisation problem with constrained second moments. This system trivially belongs to the class of systems discussed in Section 2.6. It is the case where $T = 1$, $R = 2$, $p_{r0} = 1$. We shall calculate, via a semidefinite programming algorithm, a lower bound on the achievable cost. We begin by characterising properties of the first and second moments of the queue lengths for each customer type. This will yield a relaxation of the achievable region of the problem. By optimising over this region, we can use mathematical programming methods to calculate a lower bound on the optimal cost of the problem. We can then use this lower bound to estimate the closeness to optimality of the heuristics under consideration. We use the potential function

$$R(t) = f(1) N_1(t) + f(2) N_2(t), \tag{2.44}$$

where $N_1(t)$ and $N_2(t)$ are the queue lengths of the two customer types at time t. The first recurrence relationship utilised is

$$\begin{aligned} E\left[R^2(\tau_{k+1}) \,|N(\tau_k)\right] &= \sum_{r=1}^{2} \lambda_r \left(R(\tau_k) + f(r)\right)^2 + \sum_{r=1}^{2} \mu_r \delta\left(B_r\{\tau_k\}\right)\left[\left(R(\tau_k) - f(r)\right)^2\right] \\ &\quad + \sum_{r=1}^{2} \mu_r \delta\left(\overline{B}_r\{\tau_k\}\right) R^2(\tau_k) \end{aligned}$$

$$\begin{aligned} &= \sum_{r=1}^{2} \lambda_r \left(R^2(\tau_k) + 2f(r) R(\tau_k) + f^2(r)\right) \\ &\quad + \sum_{r=1}^{2} \mu_r \delta\left(B_r\{\tau_k\}\right)\left[R^2(\tau_k) - 2f(r) R(\tau_k) + f^2(r)\right] \\ &\quad + \sum_{r=1}^{2} \mu_r \delta\left(\overline{B}_r\{\tau_k\}\right) R^2(\tau_k). \end{aligned}$$

Taking expectations on both sides gives us:

$$\begin{aligned} &E\left[E\left[R^2(\tau_{k+1}) \,|N(\tau_k)\right]\right] = \\ &\quad \sum_{r=1}^{2} \lambda_r \left\{E\left[R^2(\tau_k)\right] + 2f(r) E\left[R(\tau_k)\right] + E\left[f^2(r)\right]\right\} \\ &\quad + \sum_{r=1}^{2} \mu_r \left\{E\left[\delta\left(B_r\{\tau_k\}\right) R^2(\tau_k)\right] - 2f(r) E\left[\delta\left(B_r\{\tau_k\}\right) R(\tau_k)\right]\right. \\ &\quad \left. + E\left[\delta\left(B_r\{\tau_k\}\right) f^2(r)\right]\right\} + \sum_{r=1}^{2} \mu_r E\left[\delta\left(\overline{B}_r\{\tau_k\}\right) R^2(\tau_k)\right] \end{aligned} \tag{2.45}$$

Now, we use the identity

$$E\left\{E\left[R^2(\tau_{k+1}) \,|N(\tau_k)\right]\right\} = E\left[R^2(\tau_{k+1})\right] = E\left[R^2(\tau_k)\right].$$

Therefore, considering the $R^2(.)$ terms in particular, we can write

$$\begin{aligned} E\left[R^2(\tau_k)\right] &= \sum_{r=1}^{2} \lambda_r E\left[R^2(\tau_k)\right] + \sum_{r=1}^{2} \mu_r E\left[\delta\left(B_r\{\tau_k\}\right) R^2(\tau_k)\right] \\ &\quad + \sum_{r=1}^{2} \mu_r E\left[\delta\left(\overline{B}_r\{\tau_k\}\right) R^2(\tau_k)\right] \end{aligned}$$

$$= \sum_{r=1}^{2} \lambda_r E\left[R^2(\tau_k)\right] + \sum_{r=1}^{2} \mu_r E\left[R^2(\tau_k)\right]$$
$$= \sum_{r=1}^{2} [\lambda_r + \mu_r] E\left[R^2(\tau_k)\right]$$

and further, because under uniformisation we have that $\sum_{r=1}^{2}[\lambda_r + \mu_r] = 1$, (2.45) may be rewritten as

$$\begin{aligned} E\left[R^2(\tau_k)\right] &= E\left[R^2(\tau_k)\right] + \sum_{r=1}^{2} \lambda_r \left\{ 2f(r) E[R(\tau_k)] + E\left[f^2(r)\right] \right\} \\ &\quad -2\sum_{r=1}^{2} \mu_r f(r) E[\delta(B_r\{\tau_k\}) R(\tau_k)] + \sum_{r=1}^{2} \mu_r E\left[\delta(B_r\{\tau_k\}) f^2(r)\right]. \end{aligned}$$

Therefore, we can equate all of the remaining terms of the equation to zero as follows:

$$\begin{aligned} 0 = &\sum_{r=1}^{2} \lambda_r \left\{ 2f(r) E[R(\tau_k)] + E\left[f^2(r)\right] \right\} \\ &-2\sum_{r=1}^{2} \mu_r f(r) E[\delta(B_r\{\tau_k\}) R(\tau_k)] + \sum_{r=1}^{2} \mu_r E\left[\delta(B_r\{\tau_k\}) f^2(r)\right]. \end{aligned}$$

Now

$$E[\delta(B_r\{\tau_k\})] = E(\text{Server is busy with a type } r \text{ customer}) = \frac{\lambda_r}{\mu_r}. \qquad (2.46)$$

We now have

$$\begin{aligned} 0 = &\ 2\sum_{r=1}^{2} \lambda_r f(r) E[R(\tau_k)] + \sum_{r=1}^{2} \lambda_r E\left[f^2(r)\right] \\ &-2\sum_{r=1}^{2} \mu_r f(r) E[\delta(B_r\{\tau_k\}) R(\tau_k)] + \sum_{r=1}^{2} \lambda_r E\left[f^2(r)\right] \end{aligned}$$

and, dividing through by 2

$$\begin{aligned} 0 = &\sum_{r=1}^{2} \lambda_r f(r) E[R(\tau_k)] - \sum_{r=1}^{2} \mu_r f(r) E[\delta(B_r\{\tau_k\}) R(\tau_k)] \\ &+\sum_{r=1}^{2} \lambda_r E\left[f^2(r)\right]. \end{aligned}$$

Rearranging, we get

$$\begin{aligned} 0 \ = \ &\lambda_1 f(1) E[R(\tau_k)] + \lambda_2 f(2) E[R(\tau_k)] \\ &-\mu_1 f(1) E[\delta(B_1\{\tau_k\} R(\tau_k))] - \mu_2 f(2) E[\delta(B_2\{\tau_k\} R(\tau_k))] \\ &+\lambda_1 f^2(1) + \lambda_2 f^2(2). \end{aligned}$$

Substituting for $R(t)$ as in (2.44), we obtain

$$\begin{aligned}
0 &= \lambda_1 f(1) E[f(1) N_1(\tau_k) + f(2) N_2(\tau_k)] + \lambda_2 f(2) E[f(1) N_1(\tau_k) + f(2) N_2(\tau_k)] \\
&\quad -\mu_1 f(1) E\left[\delta(B_1\{\tau_k\})\{f(1) N_1(\tau_k) + f(2) N_2(\tau_k)\}\right] \\
&\quad -\mu_2 f(2) E\left[\delta(B_2\{\tau_k\})\{f(1) N_1(\tau_k) + f(2) N_2(\tau_k)\}\right] \\
&\quad +\lambda_1 f^2(1) + \lambda_2 f^2(2).
\end{aligned}$$

We rewrite this using the notation in (2.47) and (2.48)

$$\begin{aligned}
0 &= \lambda_1 f^2(1) n_1 - \mu_1 f^2(1) I_{11} + \lambda_1 f^2(1) + \lambda_2 f^2(2) n_2 - \mu_2 f^2(2) I_{22} + \lambda_2 f^2(2) \\
&\quad +\lambda_1 f(1) f(2) n_2 + \lambda_2 f(1) f(2) n_1 - \mu_1 f(1) f(2) I_{12} - \mu_2 f(1) f(2) I_{21}
\end{aligned}$$

where

$$n_r = E[N_r(\tau_k)] \tag{2.47}$$

$$I_{r,s} = E[\delta(B_r\{\tau_k\}) N_s(\tau_k)]. \tag{2.48}$$

Finally, equating the coefficients of $f^2(1)$, $f(1) f(2)$ etc. gives us the following sets of equations:

$$\lambda_1 n_1 - \mu_1 I_{11} + \lambda_1 = 0 \tag{2.49}$$

$$\lambda_2 n_2 - \mu_2 I_{22} + \lambda_2 = 0 \tag{2.50}$$

$$\lambda_1 n_2 + \lambda_2 n_1 - \mu_1 I_{12} - \mu_2 I_{21} = 0. \tag{2.51}$$

For the higher order interactions as described in Section 2.6.1, we use the following recursion, again using the potential function from (2.44):

$$\begin{aligned}
E\left[R^3(\tau_{k+1}) | N(\tau_k)\right] &= \sum_{r=1}^{2} \Big\{ \lambda_r (R(\tau_k) + f(r))^3 + \mu_r \delta(B_r\{\tau_k\}) \left[(R(\tau_k) - f(r))^3\right] \\
&\quad + \mu_r \delta\left(\overline{B}_r\{\tau_k\}\right) R^3(\tau_k) \Big\} \\
&= \sum_{r=1}^{2} \Big\{ \lambda_r \left(R^3(\tau_k) + 3f(r) R^2(\tau_k) + 3f^2(r) R(\tau_k) + f^3(r)\right) \\
&\quad + \mu_r \delta(B_r\{\tau_k\}) \left[R^3(\tau_k) - 3f(r) R^2(\tau_k) + 3f^2(r) R(\tau_k) - f^3(r)\right] \\
&\quad + \mu_r \delta\left(\overline{B}_r\{\tau_k\}\right) R^3(\tau_k) \Big\}.
\end{aligned}$$

Taking expectations on both sides, gives us

$$\begin{aligned} E\left[E\left[R^3(\tau_{k+1})\,|N(\tau_k)\right]\right] &= \sum_{r=1}^{2} E\Big\{\lambda_r\left(R^3(\tau_k)+3f(r)R^2(\tau_k)+3f^2(r)R(\tau_k)+f^3(r)\right) \\ &\quad +\mu_r\delta\left(B_r\{\tau_k\}\right)\left[R^3(\tau_k)-3f(r)R^2(\tau_k)+3f^2(r)R(\tau_k)-f^3(r)\right] \\ &\quad +\mu_r\delta\left(\overline{B}_r\{\tau_k\}\right)R^3(\tau_k)\Big\}. \end{aligned} \tag{2.52}$$

Now, we use the identity

$$E\Big\{E\left[R^3(\tau_{k+1})\,|N(\tau_k)\right]\Big\} = E\left[R^3(\tau_{k+1})\right] = E\left[R^3(\tau_k)\right].$$

Therefore, considering the $R^3(.)$ terms in particular, we can write

$$\begin{aligned} E\left[R^3(\tau_k)\right] &= \sum_{r=1}^{2}\Big\{\lambda_r E\left[R^3(\tau_k)\right]+\mu_r E\left[\delta\left(B_r\{\tau_k\}\right)R^3(\tau_k)\right]+\mu_r E\left[\delta\left(\overline{B}_r\{\tau_k\}\right)R^3(\tau_k)\right]\Big\}, \\ &= \sum_{r=1}^{2}\Big\{\lambda_r E\left[R^3(\tau_k)\right]+\mu_r E\left[R^3(\tau_k)\right]\Big\} \\ &= \sum_{r=1}^{2}\left[\lambda_r+\mu_r\right]E\left[R^3(\tau_k)\right] \end{aligned}$$

and, as $\sum_{r=1}^{m}[\lambda_r+\mu_r]=1$, (2.52) may be rewritten as follows:

$$\begin{aligned} E\left[R^3(\tau_k)\right] &= E\left[R^3(\tau_k)\right]+\sum_{r=1}^{2}E\left\{\lambda_r\left(3f(r)R^2(\tau_k)+3f^2(r)R(\tau_k)+f^3(r)\right)\right. \\ &\quad \left.+\mu_r\delta\left(B_r\{\tau_k\}\right)\left(-3f(r)R^2(\tau_k)+3f^2(r)R(\tau_k)-f^3(r)\right)\right\}. \end{aligned}$$

Therefore, we infer that

$$\begin{aligned} 0 = &\sum_{r=1}^{2}E\Big\{\lambda_r\left(3f(r)R^2(\tau_k)+3f^2(r)R(\tau_k)+f^3(r)\right) \\ &+\mu_r\delta\left(B_r\{\tau_k\}\right)\left(-3f(r)R^2(\tau_k)+3f^2(r)R(\tau_k)-f^3(r)\right)\Big\}. \end{aligned} \tag{2.53}$$

Utilising (2.46), we obtain that

$$\sum_{r=1}^{2}\mu E\left[\delta\left(B_r\{\tau_k\}\right)\right]f^3(r) = \sum_{r=1}^{2}\mu_r\frac{\lambda_r}{\mu_r}f^3(r) = \sum_{r=1}^{2}\lambda_r f^3(r).$$

Hence, we can rewrite (2.53) as

$$\begin{aligned} 0 = &\sum_{r=1}^{2}E\Big\{3\lambda_r f(r)R^2(\tau_k)+3\lambda_r f^2(r)R(\tau_k)+\lambda_r f^3(r) \\ &-3\mu_r\delta\left(B_r\{\tau_k\}\right)f(r)R^2(\tau_k)+3\mu_r\delta\left(B_r\{\tau_k\}\right)f^2(r)R(\tau_k)-\lambda_r f^3(r)\Big\} \end{aligned}$$

$$= \sum_{r=1}^{2} E\Big\{3\lambda_r f(r) R^2(\tau_k) + 3\lambda_r f^2(r) R(\tau_k) - 3\mu_r \delta(B_r\{\tau_k\}) f(r) R^2(\tau_k)$$
$$+3\mu_r \delta(B_r\{\tau_k\}) f^2(r) R(\tau_k)\Big\}.$$

Expanding the above and removing a factor of three, we have

$$\begin{aligned} 0 = \; & \sum_{r=1}^{2} E\Big\{\lambda_r f(r)\{f^2(1) N_1^2(\tau_k) + f^2(2) N_2^2(\tau_k) + 2f(1) f(2) N_1(\tau_k) N_2(\tau_k)\} \\ & +\lambda_r f^2(r)\{f(1) N_1(\tau_k) + f(2) N_2(\tau_k)\} \\ & -\mu_r f(r) \delta(B_r\{\tau_k\})\{f^2(1) N_1^2(\tau_k) + f^2(2) N_2^2(\tau_k) + 2f(1) f(2) N_1(\tau_k) N_2(\tau_k)\} \\ & +\mu_r f^2(r) \delta(B_r\{\tau_k\})\{f(1) N_1(\tau_k) + f(2) N_2(\tau_k)\}\Big\} \end{aligned}$$

which, on further expansion, gives

$$\begin{aligned} 0 = \; & E\Big\{\lambda_1 f(1)\{f^2(1) N_1^2(\tau_k) + f^2(2) N_2^2(\tau_k) + 2f(1) f(2) N_1(\tau_k) N_2(\tau_k)\} \\ & +\lambda_2 f(2)\{f^2(1) N_1^2(\tau_k) + f^2(2) N_2^2(\tau_k) + 2f(1) f(2) N_1(\tau_k) N_2(\tau_k)\} \\ & +\lambda_1 f^2(1)(f(1) N_1(\tau_k) + f(2) N_2(\tau_k)) + \lambda_2 f^2(2)(f(1) N_1(\tau_k) + f(2) N_2(\tau_k)) \\ & -\mu_1 f(1) \delta(B_1\{\tau_k\})\{f^2(1) N_1^2(\tau_k) + f^2(2) N_2^2(\tau_k) + 2f(1) f(2) N_1(\tau_k) N_2(\tau_k)\} \\ & -\mu_2 f(2) \delta(B_2\{\tau_k\})\{f^2(1) N_1^2(\tau_k) + f^2(2) N_2^2(\tau_k) + 2f(1) f(2) N_1(\tau_k) N_2(\tau_k)\} \\ & +\mu_1 f^2(1) \delta(B_1\{\tau_k\})(f(1) N_1(\tau_k) + f(2) N_2(\tau_k)) \\ & +\mu_2 f^2(2) \delta(B_2\{\tau_k\})(f(1) N_1(\tau_k) + f(2) N_2(\tau_k))\Big\}. \end{aligned} \tag{2.54}$$

We now use the following notation in (2.54), namely

$$\begin{aligned} n_r &= E[N_r(\tau_k)] \\ M_{rs} &= E[N_r(\tau_k) N_s(\tau_k)] \\ I_{rs} &= E[\delta(B_r\{\tau_k\}) N_s(\tau_k)] \\ H_{rsw} &= E[\delta(B_r\{\tau_k\}) N_s(\tau_k) N_w(\tau_k)] \text{ where } r, s, w = 1, 2 \end{aligned}$$

We infer that

$$\begin{aligned}0 = &\ f^3(1)(\lambda_1 M_{11} + \mu_1 I_{11} + \lambda_1 n_1 - \mu_1 H_{111}) + f^3(2)(\lambda_2 M_{22} + \mu_2 I_{22} + \lambda_2 n_2 - \mu_2 H_{222}) \\ &+ f^2(1) f(2)(2\lambda_1 M_{12} - 2\mu_1 H_{112} - \mu_2 H_{211} + \mu_1 I_{12} + \lambda_1 n_2 + \lambda_2 M_{11}) \\ &+ f^2(2) f(1)(2\lambda_2 M_{21} - 2\mu_2 H_{221} - \mu_1 H_{122} + \mu_2 I_{21} + \lambda_2 n_1 + \lambda_1 M_{22}).\end{aligned}$$

Finally, equating the coefficients of $f(1)$, $f(2)$, $f^2(1)$ and $f^2(2)$ we have the following set of constraints

$$\lambda_1 M_{11} + \mu_1 I_{11} + \lambda_1 n_1 - \mu_1 H_{111} = 0 \tag{2.55}$$

$$\lambda_2 M_{22} + \mu_2 I_{22} + \lambda_2 n_2 - \mu_2 H_{222} = 0 \tag{2.56}$$

$$2\lambda_1 M_{12} - 2\mu_1 H_{112} - \mu_2 H_{211} + \mu_1 I_{12} + \lambda_1 n_2 + \lambda_2 M_{11} = 0 \tag{2.57}$$

$$2\lambda_2 M_{21} - 2\mu_2 H_{221} - \mu_1 H_{122} + \mu_2 I_{21} + \lambda_2 n_1 + \lambda_1 M_{22} = 0. \tag{2.58}$$

The sets of equations (2.49)-(2.51) and (2.55)-(2.58) are used to define the relaxation of the achievable region.

2.7.1 Optimisation over the achievable region.

As we shall see, the sets derived using the potential function method approximate the achievable performance space (achievable region) tightly. Therefore, we are now able to solve not the optimisation problem over the exact achievable space but the corresponding problem over a (third order) relaxation of the exact space, given by $\bigcap_{l=2}^{3} P_l$. The optimal scheduling policy over the latter will give a lower bound on the optimal cost for the problem.

We now consider problems concerning a two customer single server network as presented in Section 2.3. We utilise the sets of equations derived by the potential function method (with $T = 1$, $R = 2$, $p_{r0} = 1$) in Section 2.7 to identify a relaxation of the performance space. We consider a number of problems in this section all of which seek to minimise a cost function,

$$c_1 n_1 + c_2 n_2,$$

$$\text{subject to } \{n_I, M_{ij}, I_{ij}, H_{ijk} \in P\}$$

where P is the relaxation of the state space. We also impose one or more of the alternative sets of constraint given in (2.59)-(2.61).

$$M_{22} \leq B_2 \tag{2.59}$$

$$M_{11} \leq B_1 \tag{2.60}$$

$$c_1 M_{11} + c_2 M_{22} \leq B \tag{2.61}$$

where B_1, B_2 and B are all values by which we wish to constrain the second moments of the queue lengths. We note here that it is not possible to constrain variances directly as their calculation involves the squares of the first moments. Following Ansell et al. (2001) we therefore adopt the approach of constraining second moments. P_2 is defined by (2.62)-(2.68) and these follow directly from (2.49)-(2.51).

$$\mu_1 I_{11} - \lambda_1 n_1 = \lambda_1 \tag{2.62}$$

$$\mu_2 I_{22} - \lambda_2 n_2 = \lambda_2 \tag{2.63}$$

$$\mu_2 I_{21} + \mu_1 I_{12} - \lambda_2 n_1 - \lambda_1 n_2 = 0 \tag{2.64}$$

$$\sum_{s=1}^{2} I_{1s} = n_1 \tag{2.65}$$

$$\sum_{r=1}^{2} I_{r2} = n_2 \tag{2.66}$$

$$I_{rs} \geq 0, \ \ r = 1, 2 \ \ s = 1, 2 \tag{2.67}$$

$$n_r \geq 0, \ \ r = 1, 2. \tag{2.68}$$

P_3 is defined by (2.69)-(2.76) and these follow directly from (2.55)-(2.58):

$$\lambda_1 n_1 + \lambda_1 M_{11} + \mu_1 I_{11} - \mu_1 H_{111} = 0 \quad (2.69)$$

$$\lambda_2 n_2 + \lambda_2 M_{22} + \mu_2 I_{22} - \mu_2 H_{222} = 0 \quad (2.70)$$

$$\lambda_1 M_{22} + \lambda_2 n_1 + 2\lambda_2 M_{12} + \mu_2 I_{21} - \mu_1 H_{122} - 2\mu_2 H_{221} = 0 \quad (2.71)$$

$$\lambda_2 M_{11} + \lambda_1 n_2 + 2\lambda_1 M_{12} + \mu_1 I_{12} - \mu_2 H_{211} - 2\mu_1 H_{112} = 0 \quad (2.72)$$

$$\sum_{r=1}^{2} H_{r12} - M_{12} = 0 \quad (2.73)$$

$$\sum_{r=1}^{2} H_{r11} - M_{11} = 0 \quad (2.74)$$

$$\sum_{r=1}^{2} H_{r22} - M_{22} = 0 \quad (2.75)$$

$$n_r, M_{rs}, I_{rs}, H_{rst} \geq 0. \quad (2.76)$$

The relaxation of the third order achievable performance space is thus characterised by the intersection of the two sets, P_2 and P_3 and to obtain a lower bound on each of the scheduling problems considered , we solve the following linear programming problem;

$$\text{minimise } c_1 n_1 + c_2 n_2, \quad (2.77)$$

$$\text{subject to} \{n_i, M_{ij}, I_{ij}, H_{ijk}\} \in P_2 \bigcap P_3 \quad (2.78)$$

and additional constraint(s) on the second moments as described in (2.59) - (2.61).

2.8 Semidefinite programming

Ansell et al. (1999) found that additional constraints to strengthen the above lower bound were necessary. It was felt that these should take the form of equations/inequalities linking the first and second moments which would help to further refine the relaxation of the achievable region. We outline the technique below. For details see Vandenberghe & Boyd (1996). Suppose we wish to minimise a linear function of variable x where $x \in \mathbb{R}^m$ is subject to a matrix inequality. The problem data are the vector $c \in \mathbb{R}^m$ and the $m + 1$

symmetric matrices $F_0, F_1, ..., F_m \in \mathbb{R}^{n \times n}$. We assume a problem of the form

$$\text{minimise } \mathbf{c}^T\mathbf{x},$$

$$\text{subject to } F(x) \succeq 0$$

where $F(x) \equiv F_0 + \sum_{i=1}^{m} x_i F_i$ and $F(x) \succeq 0$ denotes the requirement that $F(x)$ be positive semidefinite. The latter implies that $\mathbf{z}^T F(x)\mathbf{z} \geq 0$, all $\mathbf{z} \in \mathbb{R}^n$. The above is the standard form of a semidefinite program, which is a form of convex optimisation problem. One special case of such problems, and of interest to us here, is the standard linear programming problem:

$$\text{minimise } \mathbf{c}^T\mathbf{x},$$

$$\text{subject to } A\mathbf{x} + \mathbf{b} \geq \mathbf{0}$$

where $A = [a_1, a_2, \ldots, a_m] \in \mathbb{R}^{n \times m}$, $\mathbf{c} \in \mathbb{R}^m$ is a vector and $\mathbf{b} \in \mathbb{R}^n$ also a vector.

We denote $\mathbf{diag}(\mathbf{v})$ as the diagonal matrix having the components of $\mathbf{v}$ on the diagonal. We can state that a vector $\mathbf{v} \geq \mathbf{0}$ if and only if the matrix $\mathbf{diag}(\mathbf{v})$ is positive semidefinite. We can now rewrite the standard linear program above as the following semidefinite programming problem:

$$\text{minimise } \mathbf{c}^T\mathbf{x},$$

$$\text{subject to } F(x) \succeq 0$$

$$\text{where } F(x) \equiv F_0 + \sum_{i=1}^{m} x_i F_i, \text{ and } F_0 = \mathbf{diag}(\mathbf{b}) \text{ and } F_i = \mathbf{diag}(\mathbf{a}_i), i = 1, 2, \ldots, m.$$

We note that $\mathbf{diag}(\mathbf{a})$ and $\mathbf{diag}(\mathbf{b})$ are of the form $\mathbf{diag}(\mathbf{v})$ above. We can strengthen the formulation of constraints obtained in the preceding sections by the addition of a set of positive semidefinite constraints. These were suggested by Bertsimas & Niño-Mora (1996) and are outlined below. These additional constraints are based on the idea that the performance measures in our problem are all moments of random variables. Bertsimas & Niño-Mora (1996) show that, if a given vector $\mathbf{z}$ and a symmetric real matrix Z satisfy the necessary and sufficient condition that $Z - \mathbf{z}\prime\mathbf{z}$ be positive semidefinite, then for some random vector ψ, $\mathbf{z} = E[\psi]$ and $\mathbf{Z} = E[\psi\psi']$. This is where $\mathbf{Z} - \mathbf{z}'\mathbf{z}$ is the covariance matrix of ψ.

In the case of the single server system, additional positive semidefinite constraints are as follows:

$$D_1 = \left\{ \begin{array}{ccc} 1 & n_1 & n_2 \\ n_1 & M_{11} & M_{12} \\ n_2 & M_{12} & M_{22} \end{array} \right\} \succeq 0 \tag{2.79}$$

$$D_2 = \left\{ \begin{array}{ccc} 1 & I_{11} & I_{12} \\ I_{11} & H_{111} & H_{112} \\ I_{12} & H_{112} & H_{122} \end{array} \right\} \succeq 0 \tag{2.80}$$

$$D_3 = \left\{ \begin{array}{ccc} 1 & I_{21} & I_{22} \\ I_{21} & H_{211} & H_{212} \\ I_{22} & H_{212} & H_{222} \end{array} \right\} \succeq 0. \tag{2.81}$$

It only remains for us to reconfigure the set $P_2 \bigcap P_3$ along with the imposed second moment constraints and the additional semidefinite constraints of (2.79)-(2.81) into the form needed for a standard semidefinite program. The non-parametric bounding method produced constraints of the form $A_1\mathbf{x} - \mathbf{b_1} = \mathbf{0}$ while the standard semidefinite programming set up requires them to be in the form $A_1\mathbf{x} - \mathbf{b_1} \geq \mathbf{0}$. Thus, in order to achieve this, we re-express the constraints as follows:

$$A_1\mathbf{x} \geq \mathbf{b_1}$$

$$-A_1\mathbf{x} \geq -\mathbf{b_1}$$

where

$$A_1 = \left\{ \begin{array}{ccccccccccccccc}
-\lambda_1 & 0 & 0 & 0 & 0 & \mu_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\lambda_2 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\lambda_2 & -\lambda_1 & 0 & 0 & 0 & 0 & \mu_1 & \mu_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\lambda_1 & 0 & \lambda_1 & 0 & 0 & \mu_1 & 0 & 0 & 0 & -\mu_1 & 0 & 0 & 0 & 0 & 0 \\
0 & \lambda_2 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & \mu_2 & 0 & 0 & 0 & 0 & 0 & -\mu_2 \\
0 & \lambda_1 & \lambda_2 & 2\lambda_1 & 0 & 0 & \mu_1 & 0 & 0 & 0 & -2\mu_1 & 0 & -\mu_2 & 0 & 0 \\
\lambda_2 & 0 & 0 & 2\lambda_2 & \lambda_1 & 0 & 0 & \mu_2 & 0 & 0 & 0 & -\mu_1 & 0 & -2\mu_2 & 0 \\
-1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1
\end{array} \right\},$$

are the constraints obtained from the non-parametric bounding method. In addition,

$$\mathbf{x}^T = (n_1, n_2, M_{11}, M_{12}, M_{22}, I_{11}, I_{12}, I_{21}, I_{22}, H_{111}, H_{112}, H_{122}, H_{211}, H_{212}, H_{222})$$

and

$$\mathbf{b_1}^T = (\lambda_1, \lambda_2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0). \tag{2.82}$$

The form which must be used for the constraints on the second moments is $-A_2\mathbf{x} \geq -\mathbf{b}_2$, where for each of the constraints in (2.59) - (2.61),the required forms are given below. For $M_{11} \leq B_1$, $M_{22} \leq B_2$,

$$A_2 = \left\{ \begin{array}{ccccccccccccccc}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array} \right\}$$

$$\mathbf{b_2}^T = (B_1, B_2). \tag{2.83}$$

For $M_{22} \leq B_2$,

$$A_2 = \left\{ \begin{array}{ccccccccccccccc}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array} \right\}$$

$$\mathbf{b_2}^T = (B_2). \tag{2.84}$$

For $c_1 M_{11} + c_2 M_{22} \leq B$,

$$A_2 = \left\{ \begin{array}{ccccccccccccccc}
0 & 0 & c_1 & 0 & c_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array} \right\}$$

$$\mathbf{b_2}^T = (B). \tag{2.85}$$

We now add one further set of constraints which are included purely to ensure that all of the fifteen resultant variables are positive. This takes the form $E \geq 0$, where E represents a 15 by 15 identity matrix.

We can now write all of the linear constraints in the required form: $A\mathbf{x} - \mathbf{b} \geq \mathbf{0}$ where

$$A = \begin{pmatrix} -A_1 \\ A_1 \\ -A_2 \\ E \end{pmatrix}$$

$$\mathbf{b}^T = \left(-\mathbf{b_1}^T, \mathbf{b_1}^T, -\mathbf{b_2}^T, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\right) \tag{2.86}$$

and are now able to formulate the problem, given the particular form of A_2 with which we are concerned, as the following semidefinite program

minimise

$c_1 n_1 + c_2 n_2$

subject to

$$\begin{pmatrix} \mathbf{diag}\,(A\mathbf{x} - \mathbf{b}) & 0 & 0 & 0 \\ 0 & D_1 & 0 & 0 \\ 0 & 0 & D_2 & 0 \\ 0 & 0 & 0 & D_3 \end{pmatrix} \succeq \mathbf{0}$$

where, given a vector $\mathbf{v} \in \mathbb{R}^n$, $\mathbf{diag}\,(\mathbf{v})$ is the diagonal matrix with the components of $\mathbf{v}$ on the diagonal.

The semidefinite program constraints for the above problem can be written in the form

$$\sum_{i=1}^{15} x_i F_i - F_0 \succeq 0 \tag{2.87}$$

and then be solved. To this end, we utilised a software package developed by Fujisawa and Kojima called the SDPA (Semidefinite Programming Algorithm). We note that when the constraints on the second moments are incorporated into the objective function in a Lagrangian fashion, the semidefinite program form becomes

$$\text{minimise}$$

$$c_1 n_1 + c_2 n_2 + c_3 n_1^2 + c_4 n_2^2$$

$$\text{subject to}$$

$$\left\{ \begin{array}{cccc} \operatorname{diag}(A\mathrm{x} - \mathrm{b}) & 0 & 0 & 0 \\ 0 & D_1 & 0 & 0 \\ 0 & 0 & D_2 & 0 \\ 0 & 0 & 0 & D_3 \end{array} \right\} \succeq 0$$

where D_1, D_2 and D_3 are as defined in (2.79)-(2.81) and

$$A = \left\{ \begin{array}{c} -A_1 \\ A_1 \\ E \end{array} \right\}$$

$$\mathrm{b}^T = \left(-\mathrm{b_1}^T, \mathrm{b_1}^T, 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\right). \tag{2.88}$$

2.9 Problems involving constraints on the second moments of queue lengths

Regarding a simple two-class M/M/1 system, Ansell et al. (1999) argued that, although a strict priority policy will result in the optimal solution of a cost minimisation problem, such policies have the undesirable property of excessive queue lengths for the lower priority customers. Their service tends not only to be poor on average, but also to be extremely variable. It was to address such problems that they considered problems in which second moments of queue lengths were constrained and analysed two families of parameterised heuristic service policies: randomised policies and threshold policies. They concluded that threshold policies outperformed randomised policies over all problems of interest. Our aim is to find out the type of problem, involving some kind of constraint on the second moment of queue length, where policies based on a linear switching curve significantly outperform threshold policies. Further, it is desirable that we find policies that perform well in relation to (a lower bound on) the optimal cost on such problems.

We begin by considering one of the cost minimisation problems first posed by Ansell et al. (1999). It concerns an M/M/1 queueing system and the problem is to minimise linear holding costs, subject to a constraint on the second moment of the lower priority customer.

2.9.1 Computations

A FORTRAN program was written in which the power-series algorithm and the epsilon algorithm were used to produce a set of solvable recursive equations from a set of balance equations. Using the program, it was possible to compute the first and second moments of the expected queue lengths of the two customer types for any linear switching curve policy. We were able to enter our chosen values for the arrival rates, service rates and also the policy parameters α and β. For the calculation of the costs for the threshold policies, $\alpha = 0$ and β ranges from $1 \rightarrow \infty$ over a discrete lattice.

In our numerical study, we considered a range of problems, all for two customer M/M/1 systems as described in Section 2.3. We considered linear cost functions of the form $C = c_1 E(N_1) + c_2 E(N_2)$ where $E(N_1)$ and $E(N_2)$ are the expected queue-lengths of type 1 and type 2 customers and $c_i \in \mathbb{Z}^+$ their respective cost rates. These cost functions are subject to second moment constraint/s as given in (2.59)-(2.61).

Our aim was to find the best performance (in terms of cost) achievable by a threshold policy and by a policy based on a linear switching curve. The methods we employed to achieve this involved the computation of an expected cost for a set of linear switching curves over a range of α's and β's. By then searching over this (α, β) grid we find the lowest cost (from those computed) meeting the required second moment constraints of the given problem. As outlined in Section 2.4, policies based on a linear switching policy are characterised by two parameters α and β representing respectively the slope and intercept of the curve. Threshold policies are a subset in which $\alpha = 0$ and β is a positive integer and are thus characterised by a single parameter which we denote here by T.

Consider a two customer type M/M/1 queueing system with type 1 and type 2 customers. Arrival rates are given by $\lambda_i = (1,5)$ and service rates $\mu_i = (3,12)$, $i = 1,2$. Table 2.1 shows under $T^{cost}_{(10,1)}$, the expected costs, $c_1E(N_1) + c_2E(N_2)$ where $c = (10,1)$, and when the policy with threshold T is applied. Similarly, $T^{E(N_i^2)}_{(10,1)}$ shows $c_1E(N_1^2)+c_2E(N_2^2)$ where $c = (10,1)$, and when the policy with threshold T is applied. (This will be referred to in Section 2.11.) T ranges from 1 to 30. The second moments of queue lengths for each customer type under each policy with threshold T are also given in the columns headed $E(N_1^2)$ and $E(N_2^2)$. If we wish to find the lowest cost ($T^{*cost}_{(10,1)}$) under a threshold policy such that $E(N_2^2) < 40$ then we can see from the table that $T^{*cost}_{(10,1)} = 11.358$ and that this is achieved under the policy where $T = 14$ (from the T values included).

Searching for the best policy based on a linear switching curve over a range of α's and β's, is clearly computationally expensive. It involves carrying out a series of searches in which we slowly narrow the ranges of α's and β's to concentrate on those regions where costs achieved by a policy are lowest. Initially, we might search over a large space e.g. β = positive integers 0 to 30 and α = 0.5 to 5.0 in steps of 0.5. This would then be progressively narrowed by searching a smaller area (where the costs are lowest) i.e. β over fewer integers and α over smaller ranges and with smaller steps, say 0.1, then 0.01.

As threshold policies are characterised by the single parameter, T, the search for the lowest threshold cost is clearly much simpler and hence requires far less processing time. Let T^* denote the value of parameter T in the threshold policy which achieves the lowest cost. Thus in the example above we have $T^* = 14$. We found by experiment that, in a given problem, it was best to first find the lowest cost under a threshold policy so that we had a value for T^*. We could then use T^* to help us define the initial range of β over which we compute costs for our set of linear switching curves (this would usually be $T^* - 4 \leq \beta \leq T^* + 4$). This helped us reduce the subsequent computation time. Note that

Table 2.1: Costs under threshold policies for a two customer type M/M/1 system where $\lambda_1 = 1, \lambda_2 = 5, \mu_1 = 3,\ \mu_2 = 12$.

T	$E(N_1^2)$	$E(N_2^2)$	$T_{(10,1)}^{E(N_i^2)}$	$T_{(10,1)}^{cost}$	T	$E(N_1^2)$	$E(N_2^2)$	$T_{(10,1)}^{E(N_i^2)}$	$T_{(10,1)}^{cost}$
1	6.627	1.735	68.003	16.428	16	2.071	43.937	64.649	11.086
2	6.026	2.986	63.249	15.656	17	1.962	46.408	66.025	10.971
3	5.489	4.983	59.871	14.979	18	1.864	48.764	67.382	10.868
4	5.010	7.501	57.599	14.391	19	1.776	50.952	68.709	10.778
5	4.584	10.368	56.204	13.881	20	1.697	53.029	69.998	10.697
6	4.204	13.459	55.502	13.437	21	1.626	54.980	71.242	10.625
7	3.867	16.679	55.345	13.048	22	1.563	56.809	72.437	10.560
8	3.566	19.958	55.615	12.707	23	1.506	58.522	73.580	10.502
9	3.297	23.241	56.215	12.408	24	1.455	60.122	74.670	10.451
10	3.058	26.487	56.974	12.143	25	1.409	61.616	75.706	10.405
11	2.844	29.665	58.108	11.910	26	1.368	63.009	76.687	10.363
12	2.653	32.753	59.286	11.703	27	1.331	64.303	77.615	10.326
13	2.482	35.734	60.560	11.520	28	1.298	65.512	78.489	10.292
14	2.330	38.597	61.897	11.358	29	1.268	66.632	79.310	10.263
15	2.193	41.332	63.266	11.214	30	1.241	67.672	80.082	10.236

index policies of Chapter 3 have the advantage that the time taken to run the programme to calculate the expected costs is greatly reduced as it involves no search procedure.

2.10 Initial Results

2.10.1 Single constraint on the second moment of the length of the lower priority queue

In our initial work, we calculate costs for integer intercepts only, in the linear switching policies, the motivation being to quickly identify whether the switching curve policy offers a significant improvement and to enable us to compare simply its performance in a number of problems where the same objective function had a variety of second moment constraints. We seek to

$$\text{minimise } C = c_1 E(N_1) + c_2 E(N_2)$$
$$\text{subject to } E\left(N_2^2\right) \leq v_2 \quad \text{where } v_2 = 74.641.$$

The figure of 74.641 was simply chosen to correspond with results from Ansell et al. (1999). The results for each of the policies are given below. $Cost_{(10,1)}^{best}$ is the lowest cost found for a given policy class (or in the case of semidefinite lower bound, SDLB, the lower bound cost based on our relaxation of the performance space) when $c_1 = 10$ and $c_2 = 1$.

Policy	$Cost_{(10,1)}^{best}$	Parameter value
Threshold	10.086	$T = 40$
L. S. Curve	10.084	$\alpha = 0.4$ $\beta = 38$
SDLB	10.000	

The policy based on a linear switching curve gave a result 0.84% above the SDLB. The threshold policy gave a result 0.86% above the SDLB and 0.02% greater than that of the switching curve policy. Even when we reduced the constraint to $v_2 \leq 32$, little improvement was achieved on the best performance by a threshold policy from within the linear switching class.

2.10.2 Constraints on both second moments.

We decided to investigate the performances of the two classes of policy in a problem where there were constraints on the second moments of *both* customer types. The problem considered was

minimise $C = c_1 E(N_1) + c_2 E(N_2)$

such that

$E(N_2^2) \leq v_2$

$E(N_1^2) \leq v_1$

where $v_2 = 32$ and $v_1 = 5$.

The results for this were broadly similar to those for the above problem with the single constraint. This was felt to be because the relatively large differences in arrival rates (Type 2 customers arrive at a rate five times that of type 1) and cost rates ($c_1 = 10$ and $c_2 = 1$) of the two customer types meant that the additional constraint, $E(N_1^2) \leq 5$, hardly impacted the results. Hence, the problem effectively had a single second moment constraint.

Policy	$Cost^{best}_{(10,1)}$	Parameter value
Threshold	11.910	$T = 11$
Linear switching curve	11.761	$\alpha = 0.42$ $\beta = 10$

This gave a best threshold cost 1.27 % greater than the best switching curve cost.

Our aim, therefore, became to find a way to constrain the problem in such a way that *both* second moments bite.

2.11 Constraining the problem by a linear sum of the second moments.

We decided to impose the constraint: $10E(N_1^2) + E(N_2^2) \leq 55.345$ as this reflected the priority shown to the type 1 customer in terms of cost. The value of 55.345 was chosen as it was the minimum value of $10E\left(N_1^2\right) + E\left(N_2^2\right)$ to be offered by any of the threshold policies in Table (2.1) (i.e. the lowest value in column $T_{(10,1)}^{E\left(N_i^2\right)}$). The results obtained from optimising within the various policy classes now became

Policy	$Cost_{(10,1)}^{best}$	Parameter value
Threshold	13.048	$T = 7$
Linear switching curve	11.841	$\alpha = 2.3$ $\beta = 5$
SDLB	10.799	

The policy based on a linear switching curve gave a result 9.649% above the SDLB. The threshold policy gave a result 20.826% above the SDLB and an increase of 10.193% over the switching curve based cost. Clearly, this is more substantial improvement and merited further investigation.

2.11.1 Problems with varying ρ

Constraining the problem by a linear sum of the second moments of the two customer types was the problem formulation for which the linear switching classes outperformed the threshold policies to the largest degree in approaching the SDLB. We therefore used this form of constraint in all of the remaining problems investigated numerically in this chapter. We continued our investigation of the performance of policies based on linear switching curves by considering the effect of varying the parameter ρ, the traffic intensity. The problems considered were of the form

$$\text{minimise } C = 10E\left(N_1\right) + E\left(N_2\right)$$
$$\text{such that } 10E\left(N_1^2\right) + E\left(N_2^2\right) \leq T_{(10,1)}^{*E\left(N_i^2\right)}$$

where $T_{(10,1)}^{*E\left(N_i^2\right)}$ is the minimum value of $10E\left(N_1^2\right) + E\left(N_2^2\right)$ achieved under any threshold policy. The arrival rates and service rates are as indicated in Table 2.2. We use the notation C^T for the cost under the threshold policy with the lowest value of $10E\left(N_1^2\right) + E\left(N_2^2\right)$; C^{SW} is the lowest cost found under our search strategy for a linear switching policy and

C^{SD} is the semidefinite lower bound on the optimum cost. $\%^{T>SD}$ is the percentage increase on the semidefinite lower bound accrued by adopting the best threshold policy, and $\%^{SW>SD}$ is the percentage increase on the semidefinite lower bound accrued by adopting the policy based on the best switching curve found.

Table 2.2: Results for systems with varying values of ρ

λ_1	λ_2	μ_1	μ_2	ρ	C^T	C^{SW}	C^{SD}	$\%^{T>SW}$	$\%^{T>SD}$	$\%^{SW>SD}$
1	5	4.0	12	0.667	6.922	6.361	5.830	8.819	18.731	9.108
1	5	3.0	12	0.750	13.049	11.840	10.799	10.211	20.835	9.640
1	4	2.5	10	0.800	18.032	16.700	15.194	7.976	18.678	9.912
1	5	2.5	12	0.817	21.964	20.264	19.284	8.389	13.897	5.082
1	5	2.5	10	0.900	38.412	33.396	31.533	15.020	21.815	5.908
1	5	2.0	12	0.917	61.463	58.307	57.393	5.413	7.091	1.593

We observe that the linear switching class offers considerable improvement in performance on threshold based policies. At the highest value of ρ, 0.917, the percentage difference between the semidefinite cost and the cost resulting from the adoption of a switching curve based policy was only 1.59%. We note that as ρ increases, so the computational time increased, as convergence took longer.

2.11.2 Problems where $\mu_1 = \mu_2 = 1$

Having allowed ρ to vary, we continued our investigations as described below but with ρ fixed at 0.75. This is a level at which the system could be said to be in moderately heavy traffic but which was not so computationally expensive as to severely restrict the number of problems we could analyse. We first consider a set of problems where both customer types are served at the same rate. In each problem, service rates, (μ_1, μ_2) for the two customer types are (1,1) and λ_1 is randomly generated from a $U(0.1\ ,\ 0.65)$ distribution while λ_2 is chosen so that $\lambda_1 + \lambda_2 = 0.75$. The results from these problems are shown in Table 2.3. It is clear that the costs offered by the threshold policies are virtually indistinguishable from the semidefinite lower bound costs and such policies are thus very close to optimal for these problems.

Table 2.3: Results for systems with $\mu_1 = 1$ and $\mu_2 = 1$

λ_1	λ_2	C^T	C^{SD}	λ_1	λ_2	C^T	C^{SD}
0.312	0.438	7.081	7.081	0.472	0.278	11.059	11.059
0.163	0.587	4.751	4.748	0.471	0.279	11.023	11.023
0.450	0.300	10.362	10.361	0.178	0.572	4.957	4.953
0.398	0.352	8.949	8.949	0.264	0.486	6.230	6.229
0.165	0.585	4.782	4.778	0.499	0.251	11.965	11.965
0.575	0.175	15.182	15.182	0.580	0.170	15.417	15.417
0.604	0.146	16.754	16.754	0.135	0.615	4.418	4.409
0.432	0.318	9.846	9.846	0.454	0.296	10.473	10.473
0.124	0.626	4.280	4.269	0.365	0.385	8.169	8.169
0.600	0.150	16.494	16.494	0.271	0.479	6.355	6.354

2.11.3 Problems where $\mu_1 = 3, \mu_2 = 12$

We decided to return to the service rates, $\mu_1 = 3$ and $\mu_2 = 12$, of our original example to complete our investigations. Again ρ is fixed at 0.75. λ_1 is randomly generated on the interval $(0, 2)$ and $\lambda_2 = 9 - 4\lambda_1$. The results are in Table 2.4 and Table 2.5.

Table 2.4: Results for systems with $\mu_1 = 3$ and $\mu_2 = 12$

λ_1	λ_2	C^T	C^{SW}	C^{SD}	$\%^{T>SW}$	$\%^{T>SD}$	$\%^{SW>SD}$
0.228	8.089	6.348	6.222	5.840	2.041	8.716	6.541
0.231	8.076	6.395	6.160	5.860	3.815	9.130	5.119
0.285	7.861	7.116	6.466	6.350	10.053	12.063	1.827
0.323	7.708	7.615	6.782	6.640	12.283	14.684	2.139
0.412	7.352	8.314	7.470	7.309	11.299	13.750	2.203
0.534	6.865	9.297	8.330	8.117	11.609	14.537	2.624
0.558	6.766	9.573	8.558	8.226	11.860	16.375	4.036
0.597	6.614	9.995	8.922	8.506	12.026	17.505	4.891
0.624	6.506	10.293	9.007	8.657	14.278	18.898	4.043
0.671	6.316	10.380	9.343	8.912	11.099	16.472	4.836
0.693	6.228	10.613	9.503	9.045	11.681	17.336	5.064
0.771	5.917	11.442	10.200	9.496	12.176	20.493	7.414
0.775	5.902	11.483	10.116	9.532	13.513	20.468	6.127
0.825	5.702	11.601	10.491	9.821	10.580	18.124	6.822
0.963	5.148	12.664	11.609	10.577	9.088	19.731	9.757
1.083	4.666	13.917	12.575	11.247	10.672	23.740	11.808
1.182	4.274	14.592	13.331	11.934	9.459	22.272	11.706
1.207	4.170	14.864	13.563	12.022	9.592	23.640	12.818
1.240	4.042	15.203	13.837	12.296	9.872	23.642	12.533
1.243	4.029	15.239	13.870	12.295	9.870	23.945	12.810

Table 2.5: Results for systems with $\mu_1 = 3$ and $\mu_2 = 12$ continued

λ_1	λ_2	C^T	C^{SW}	C^{SD}	$\%^{T>SW}$	$\%^{T>SD}$	$\%^{SW>SD}$
1.248	4.007	14.977	13.872	12.294	7.966	21.824	12.836
1.272	3.910	15.234	14.108	12.498	7.981	21.892	12.882
1.286	3.856	15.379	14.292	12.645	7.606	21.621	13.025
1.350	3.599	16.068	14.788	13.306	8.656	20.758	11.138
1.354	3.583	16.112	14.852	13.348	8.484	20.707	11.268
1.393	3.427	16.254	15.202	13.770	6.920	18.039	10.399
1.412	3.352	16.458	15.359	13.983	7.155	17.700	9.841
1.451	3.196	16.890	15.810	14.421	6.831	17.121	9.632
1.726	2.096	19.534	18.772	18.033	4.059	8.324	4.098
1.728	2.089	19.556	18.796	18.070	4.043	8.224	4.018
1.730	2.078	19.593	18.834	18.086	4.030	8.332	4.136
1.736	2.055	19.667	18.894	18.181	4.091	8.173	3.922
1.745	2.022	19.779	19.029	18.330	3.941	7.905	3.813
1.818	1.729	20.645	20.024	19.504	3.101	5.850	2.666
1.834	1.662	20.885	20.267	19.766	3.049	5.661	2.535
1.838	1.648	20.938	20.309	19.842	3.097	5.524	2.354

$\%^{T>SW}$ represents percentage increase on the cost incurred by following the best threshold policy instead of the policy based on the best switching curve found. Values range from 2.041 to 14.278 with median 8.872.

$\%^{T>SD}$ ranges from 5.524 to 23.945 with median 17.420

$\%^{SW>SD}$ ranges from 1.827 to 13.025 with median 5.623. It is clear that the threshold policies are considerably outperformed in every case and that the switching curve based policies are able to approach the semidefinite cost more closely.

2.12 Conclusion

In our numerical investigations into problems with quadratic cost constraints, we have made progress in assessing both the relative performances of threshold policies and those based on linear switching curves and their absolute performance as measured against a theoretical lower bound. Given a two class M/M/1 system with moderately heavy traffic those policies based on a linear switching curve did perform well. Threshold policies were shown to be close to optimal in those cases where the service rates of the two customer classes were equal.

The techniques employed, however, were computationally expensive in that we were obliged to search exhaustively for a switching curve which offered the best performance. Motivated by these considerations, in the following chapters, we derive index based policies for n customer classes which, we shall show by numerical investigations, perform well for two and three customer type M/M/1 systems. In the two customer system with quadratic costs as part of the objective function, the index policy will take the form of a linear switching curve. We are able to use the index to simply calculate the best (close to best) performing switching curve, thus removing the need for general searches to be undertaken.

Chapter 3

Whittle index Policies

3.1 Introduction

In Chapter 2 we showed that policies based on a linear switching curve were able to outperform threshold policies and were close to optimal for problems where the cost function was a linear combination of the expected queue lengths and constrained by a linear combination of second moments of queue lengths. There were, however, limitations in this: it was necessary to run a search for the values of the two policy parameters α and β to minimise the given cost function. Even with the simple single server two customer type system of Chapter 2, this is time consuming. If we wish to extend our model to n customer types then the amount of processor time would quickly become prohibitive. In this chapter, and the one which follows, we consider the problem of how best to allocate a single server in an M/M/1 system among the queues of K waiting customer classes in order to minimise costs when the system cost rate is increasing convex in the number present within each class. Thus we assume that the marginal increase in the system cost rate which results from one extra customer increases with the number already present. Essentially, the model is kept as simple as possible. We assume that Markovian dynamics operate within the system, with new customers arriving in independent Poisson streams. All service rates are assumed to be exponentially distributed and independently, identically distributed within each customer class.

As the cost function of the system we consider is non-linear, our problem cannot be analysed using the classical theory of Gittins indices. It does however, bear close comparison to a variant of the multi-armed bandit problem, namely the restless bandit problem. The distinguishing feature of the restless bandit problem is that projects competing for the attention of the server may change state even when no processing time is allotted to them.

Whittle (1988) introduced this class of problem and proposed an index-based approach for their solution in which the index for each project/customer type depends on its current state. He also considered *index policies* where service is allocated to the project with the largest index value. However, for restless bandit problems in general, Whittle's indices do not necessarily exist, nor are index policies necessarily optimal. Thus it is first necessary to address the issue of the *indexability* of the system in any analysis of a restless bandit problem. Current knowledge of when such indices do exist is incomplete. Niño-Mora (2001*b*) has advanced the work by expounding a set of conditions sufficient for project indexability. Further, Weber & Weiss (1990) and Weber & Weiss (1991) have shown that, given certain conditions, index policies offer a form of asymptotic optimality.

In this chapter, we prove the indexability of an important class of discounted costs queueing control problems, demonstrate by means of simple arguments, indices in closed form and then go on to develop indices for the average cost version of the problem.

3.2 The Model

We consider a system with K customer classes, labelled $\{1, ..., K\}$. Customers arrive for service in independent Poisson streams where λ_k signifies the rate for class k. Each class k customer has a processing requirement or service time which is exponentially distributed with rate μ_k, where $1 \leq k \leq K$. On service completion, a customer leaves the system. All inter-arrival times and service times are assumed to be independent.

At each decision epoch, the system controller must decide which of the waiting customers should be served next in order to minimise some measure of expected holding cost. Decisions epochs occur on the arrival of any new customer and whenever a service completion results in a non-empty system. Thus, for example, if a class k customer enters service at time t then the next decision epoch will occur at time $t + X$, where $X \sim exp\left(\mu_k + \sum_{j=1}^{K} \lambda_j\right)$. We also make the standard assumption $\rho = \sum_{j=1}^{K} \frac{\lambda_j}{\mu_j} < 1$ to ensure finite queue lengths.

The state of the system at time t is represented by the vector of queue lengths,

$$\mathrm{N}(t) = \{N_1(t), N_2(t), ..., N_K(t)\}, \qquad t \in \mathbb{N}. \tag{3.1}$$

If we denote by a_k the action of allocating service to a class k customer, then, at each decision epoch, the controller selects an action a_k from the set of K, for which $N_k(t) \geq 1$. We use the following notation; $\Lambda = \sum_{k=1}^{K} (\lambda_k + \mu_k)$. We also use standard uniformisation in which successive decision epochs occur at the event times of a Poisson process with rate Λ and where events corresponding to service being offered are virtual state transitions. Thus, for example, if the system is in state $\mathrm{N}(t) = \mathbf{n}$ where $n_k > 0$, and action a_k is taken at time t, then the next decision epoch occurs at time $t + X$ where $X \sim exp(\Lambda)$. The system state after any state transition is described below:

$$\mathrm{N}\left\{(t+X)^+\right\} = \begin{cases} \mathbf{n} + \mathbf{1}^j & \text{with probability } \lambda_j/\Lambda, 1 \leq j \leq K, \\ \mathbf{n} - \mathbf{1}^k & \text{with probability } \mu_k/\Lambda, \\ \mathbf{n} & \text{with probability } \sum_{j \neq k} \mu_k/\Lambda. \end{cases}$$

Between t and $t + X$, the system incurs discounted costs at rate

$$\alpha \sum_{l=1}^{K} C_l(n_l) \tag{3.2}$$

where the functions C_l, satisfy the conditions set out below. We aim to minimise a measure of expected holding cost. We note that the α is necessary for the discounted costs version of the problem to guarantee that Lemma 3.1 holds. We assume that the class l holding cost rate function, $C_l : \mathbb{N} \rightarrow \mathbb{R}^+$ and $C_l(0) = 0$ is

- increasing,
- convex,
- bounded above by a polynomial of finite order (guaranteeing that all required expectations exist).

We also assume that server control is

- non-anticipative,
- non-idling,
- pre-emptive.

We assume there are no cost penalties when the server switches between customers and switches of service are considered instantaneous. The class of admissible controls is denoted by U.

We consider a stochastic optimisation problem with discounted costs described in (3.3)

$$C(\mathbf{n}, \alpha) = \inf_{u \in U} E_u \left[\int_0^{\infty} \sum_{k=1}^{K} C_k \{ N_k(t) \} \alpha e^{-\alpha t} dt \mid N(0) = \mathbf{n} \right]. \quad (3.3)$$

$C(\mathbf{n}, \alpha)$ is the minimum system cost incurred when the system is operated from time 0 with initial state $\mathbf{n}$. $N_k(t)$ is the number of k class customers present in the system at time t. E_u denotes an expectation taken over all realisations of the system under policy u and $\alpha > 0$ is a discount rate.

The related stochastic optimisation problem of primary interest is to determine the minimum cost and to identify a policy by which this cost is achieved. This can be formally described as

$$C^{OPT} = \inf_{u \in U} \tilde{E}_u \left\{ \sum_{k=1}^{K} C_k(N_k) \right\} \quad (3.4)$$

where N_k denotes the number of class k customers in the system and $\tilde{E}_u$ is an expectation with respect to the system in steady state under policy u.

Lemma 3.1 follows from standard results of dynamic programming and shows how (3.3) and (3.4) are related.

Lemma 3.1 *For all initial states* $\mathbf{n}$,

$$\lim_{\alpha \to 0} C(\mathbf{n}, \alpha) = C^{OPT}.$$

In light of Lemma 3.1, we develop policies which perform well for the average cost problem in (3.4) as limits, when $\alpha \to 0$, of policies which perform well for the discounted costs problems of (3.3). It is the latter which will be our starting point.

The classical approach to finding u^{OPT}, a policy which minimises the costs in (3.3), would utilise the techniques of stochastic dynamic programming. The employment of such techniques in this case is unlikely to yield insights into the reasons why and how such a policy actually does yield the minimum cost and consequently would be unlikely to be of assistance in extending future work to the solution of more general problems. Even for the problem considered here, there is also the curse of dimensionality, which is an issue for large K. We therefore seek *heuristic policies* which perform well in that they are simply structured and close to cost minimising.

Following the ground-breaking work by Whittle (1988) on the restless bandit problem, we concentrate our efforts on *index policies.* We want to identify class-specific index functions, $W_{k,\alpha} : \mathbb{Z}^+ \to \mathbb{R}, 1 \leq k \leq K$, such that the policy which, at each decision epoch, chooses to allocate the server to the non-empty queue with the greatest index value, $W_{k,\alpha}\{N_k(t)\}$ is close to optimal.

3.3 Indexability and Whittle indices

In this section, we aim to identify class-specific index functions as described in the previous section above. We require that each index is a function only of the stochastic dynamics and cost structure of the class concerned and the Whittle index is as such. Bertsimas & Niño-Mora (1996) refer to this as *decomposability.* It is precisely because of this property of decomposability that we are able to continue our quest to identify a Whittle index for our discounted cost problem by restricting our attention to a *single customer class.* We therefore proceed to drop the class identifier, k, and now simply denote the class index by $W_\alpha(n)$ when there are n customers, (of the single class with which we are concerned) present in the system. We develop $W_\alpha(n)$ as a *subsidy for passivity* of a customer class when n customers of that class are present in the system. An alternative approach, yielding exactly the same result, would be to regard $W_\alpha(n)$ as a charge for activity when n customers of that class are present in the system.

We consider a Markov Decision Problem with a single customer class and state space $\mathbb{N}$. At each decision epoch (the arrival of a new customer or a service completion) a choice is made as to whether the server is to be switched on or not. Thus in each state there are two possible actions for the server: $\{$active, passive$\}$. Clearly in state 0, only the passive action is possible. When the server is active, a customer is served and has a processing requirement which is exponentially distributed with rate μ. Assuming the system is in state n and active, then it either enters state $(n+1, n-1)$ with rates λ and μ respectively. When the system is in passive mode, it is frozen until a new customer arrives and hence enters state $n+1$ from n at rate λ.

Costs are assumed to be incurred at a discounted rate of $\alpha C(n)$ under the active action and $\alpha C(n) - W$ under the passive action. W is a subsidy for passivity. The optimisation problem with which we are concerned involves finding a policy for switching the server on and off so that we minimise the total holding costs and passive subsidies incurred over an infinite time horizon. This MDP is a *restless bandit.*

$C(n, \alpha, W)$ denotes this minimised cost when the initial state of the system is $n \in N$. We write

$$C(n,\alpha,W) = \min_{u \in \bar{U}} E_u \left\{ \int_0^\infty [\alpha C\{N(t)\} - WI(t)]\, e^{-\alpha t} dt \mid N(0) = n \right\} \tag{3.5}$$

where $N(t)$ is the number of customers in the system at time t, $I(t)$ is an indicator which is 1 when the server is off/passive at time t and 0 otherwise and $\bar{U}$ is the class of stationary policies for the problem.

By standard theory, the function $C(., \alpha, W)$ satisfies the optimality equations:

$$C(n,\alpha,W) = \min\{C_1(n,\alpha,W), C_2(n,\alpha,W)\}, n \in \mathbb{Z}^+ \tag{3.6}$$

where

$$(\alpha+\lambda+\mu)\,C_1(n,\alpha,W)=\alpha C(n)+\mu C(n-1,\alpha,W)+\lambda C(n+1,\alpha,W) \tag{3.7}$$

and

$$(\alpha+\lambda+\mu)\,C_2(n,\alpha,W)=\alpha C(n)-W+\mu C(n,\alpha,W)+\lambda C(n+1,\alpha,W). \tag{3.8}$$

Also, $C(0,\alpha,W)=C_2(0,\alpha,W)$ so we can write

$$(\alpha+\lambda)\,C(0,\alpha,W)=\lambda C(1,\alpha,W).$$

Equations (3.7) and (3.8) are respectively the results of choosing the active or passive action in the initial state n. If $C_1(n,\alpha,W)\leq C_2(n,\alpha,W)$ then the active action is optimal and if $C_2(n,\alpha,W)\leq C_1(n,\alpha,W)$ then the passive action is optimal. It is important to note that when $C_1(n,\alpha,W)=C_2(n,\alpha,W)$ then *both* the active and passive actions are optimal.

We use the term $\Pi_\alpha(W)$ to denote the set of states in which it is optimal to choose the *passive* action when the reward for passivity is W or, more formally

$$\Pi_\alpha(W)=\{0\}\bigcup\{n\in\mathbb{Z}^+;C_2(n,\alpha,W)\leq C_1(n,\alpha,W)\}\,,\quad W\in\mathbb{R}^+.$$

The following defines the notion of indexability for an individual class as developed by Whittle (1988)

Definition 3.1 *The class is indexable if* $\Pi_\alpha:\mathbb{R}\to 2^{\mathrm{N}}$ *is increasing, i.e.*

$$W_1>W_2\Rightarrow\Pi_\alpha(W_1)\supseteq\Pi_\alpha(W_2).$$

Once the notion of indexability is established, there follows thereby the notion of a *state* n *index* as the minimum subsidy for passivity for which the passive action is optimal in state n.

Definition 3.2 *When the class is indexable, the* Whittle index *for state n is given by:*

$$W_\alpha(n)=\inf\{W;n\in\Pi_\alpha(W)\}\,,n\in\mathbb{Z}^+, \tag{3.9}$$

where

$$W_\alpha(0) = 0. \tag{3.10}$$

Lemma 3.2 follows trivially from the above:

Lemma 3.2 *For all states n of an indexable class*

$$W \geq W_\alpha(n) \Rightarrow \textit{the passive action is optimal;} \tag{3.11}$$

$$W \leq W_\alpha(n) \Rightarrow \textit{the active action is optimal.} \tag{3.12}$$

This is illustrated in Fig. 3.1.

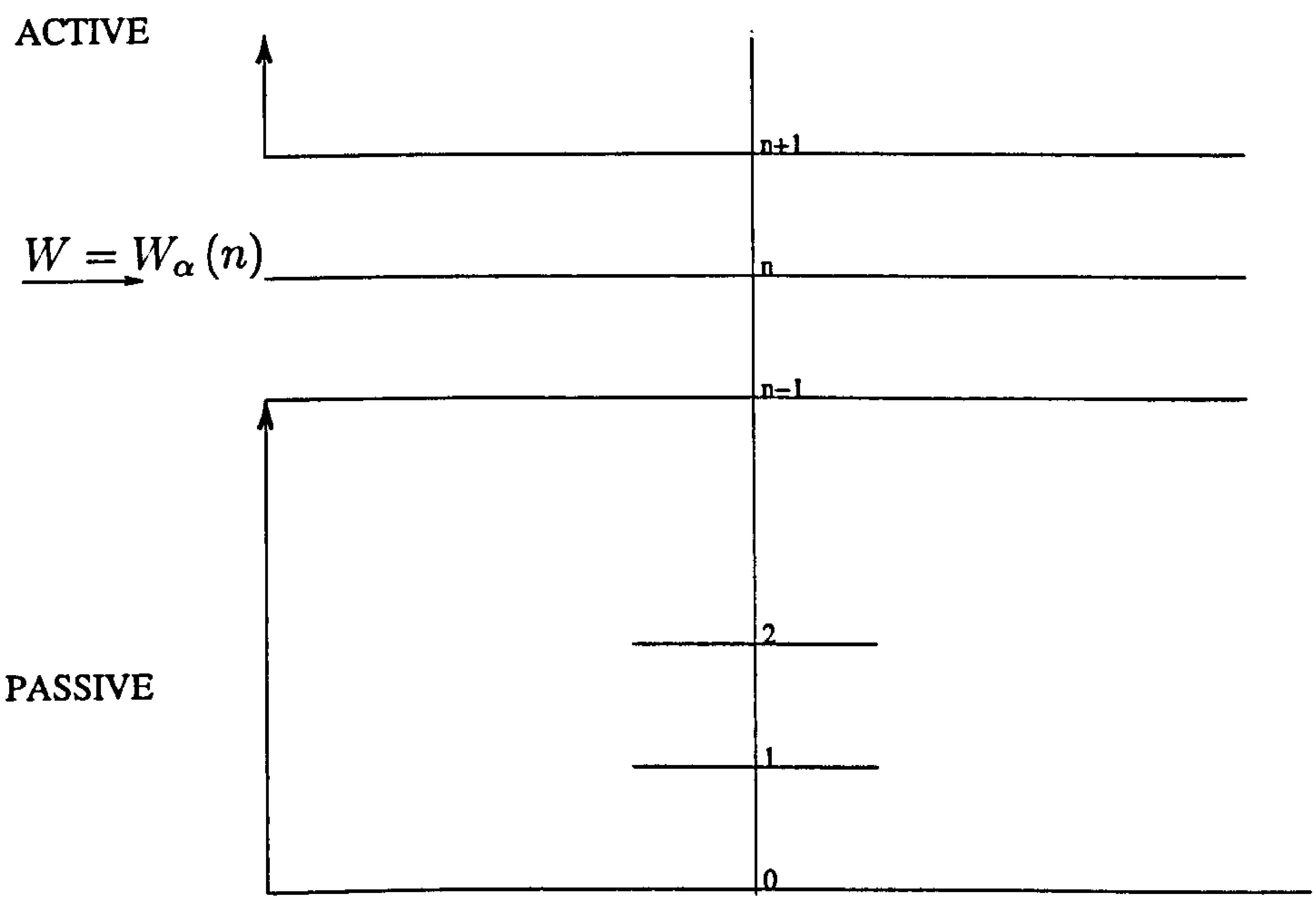

Figure 3.1: Optimal actions for states when $W = W_\alpha(n)$

We now consider the single class problem in initial state n. We suppose for now that the class is indeed indexable and that the Whittle index $W_\alpha : \mathbb{N} \rightarrow \mathbb{Z}^+$ is increasing. The subsidy for passivity is taken to be $W = \bar{W}_\alpha(n)$ where $\bar{W}_\alpha(n)$ is the *assumed* value of the index. Thus, for the optimal policy, the following will hold if $\bar{W}_\alpha$ is assumed to be increasing in n as seems reasonable.

(i) The active option will be optimal for states $\{n+1, n+2, ...\}$

(ii) The passive option will be optimal for states $\{0, ..., n-1\}$

(iii) Both the active *and* passive actions will be optimal for state n.

In (iii) we can choose the active or passive action for the server (as both actions are optimal in state n). We now consider each of these possible actions in turn and, by use of an heuristic argument, we develop the form of the index.

3.3.1 Active action in n

First we consider the restless bandit determined by the choice of the active option in n, i.e. under the stationary policy which chooses,

(i) active for states $\{n, n+1, ...\}$

(ii) passive for states $\{0, ..., n-1\}$.

The system evolution starting from initial state n can be described as follows. We begin at time 0 in state n and so the server is active. The active action continues until the system enters the state $n-1$ for the first time, at time T. The length of time that the initial active period actually lasts, T, is a random variable and is stochastically identical to the busy period of an M/M/1 queue starting with one customer, arrival rate λ and service rate μ.

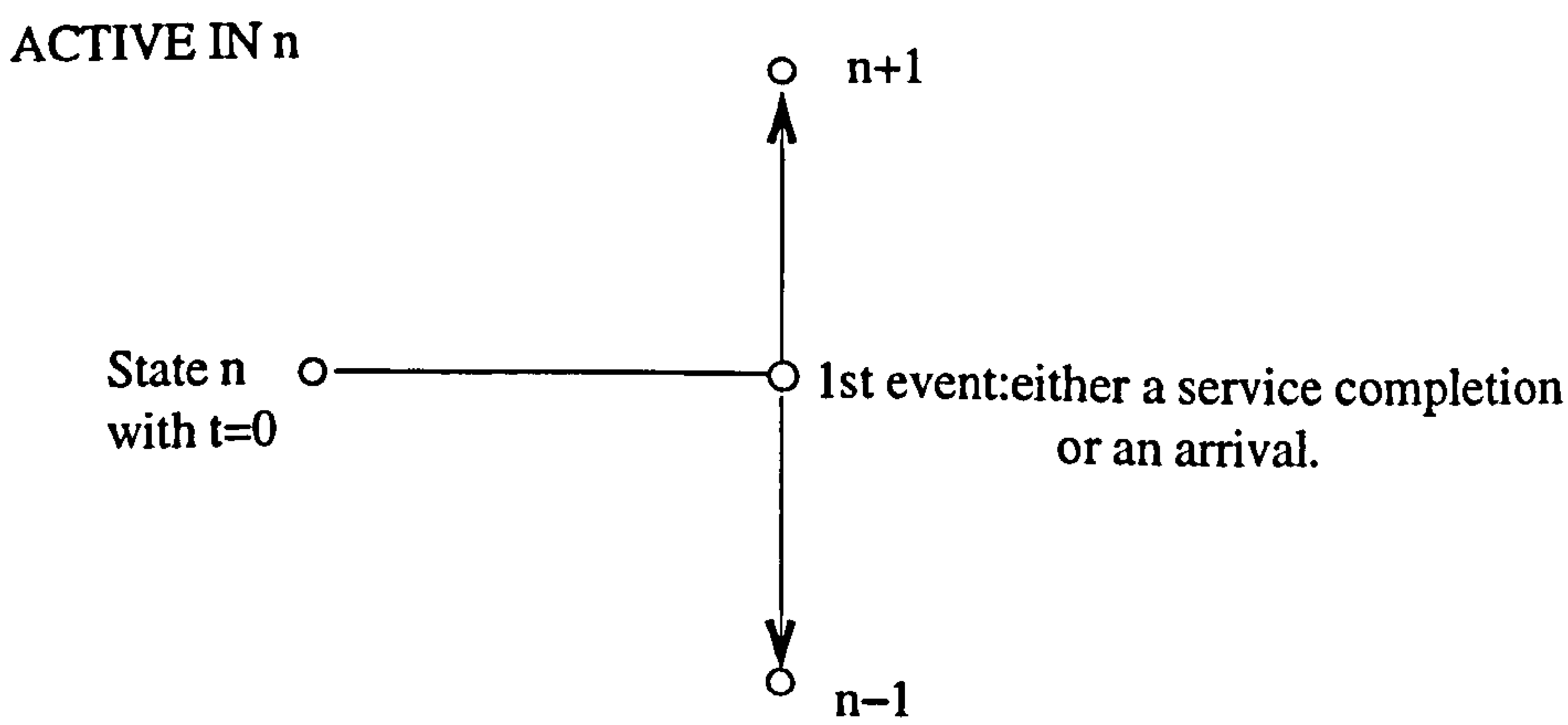

Figure 3.2: The length of the first active service period.

We think of the busy period as beginning at the time $t = 0$ with the start of service of customer n. The first event will either be a departure, in which case the busy period ends as there are now $n-1$ customers in the system or an arrival.(See Fig 3.2.) If it is an arrival, then we have in effect two busy periods to complete; $n+1$ to n and n to $n-1$. Hence, conditioning on the nature of this first event, we obtain;

$$E\left(e^{-\alpha T}\right) = \left(\frac{\lambda+\mu}{\alpha+\lambda+\mu}\right)\left(\frac{\mu}{\lambda+\mu}\right) + \left(\frac{\lambda+\mu}{\alpha+\lambda+\mu}\right)\left(\frac{\lambda}{\lambda+\mu}\right) E\left(e^{-\alpha T}\right)^2, \quad (3.13)$$

or

$$\lambda\{E\left(e^{-\alpha T}\right)\}^2 - (\alpha+\lambda+\mu)\, E\left(e^{-\alpha T}\right) + \mu = 0. \quad (3.14)$$

At time T, the system enters the state $n-1$, therefore the passive action is optimal and will remain so until the system returns to state n. The length of this passive period is also a random variable, which is exponentially distributed with rate λ. Note that when the system has returned to state n then the above cycle repeats itself ad infinitum. This is illustrated below in Fig 3.3 below.

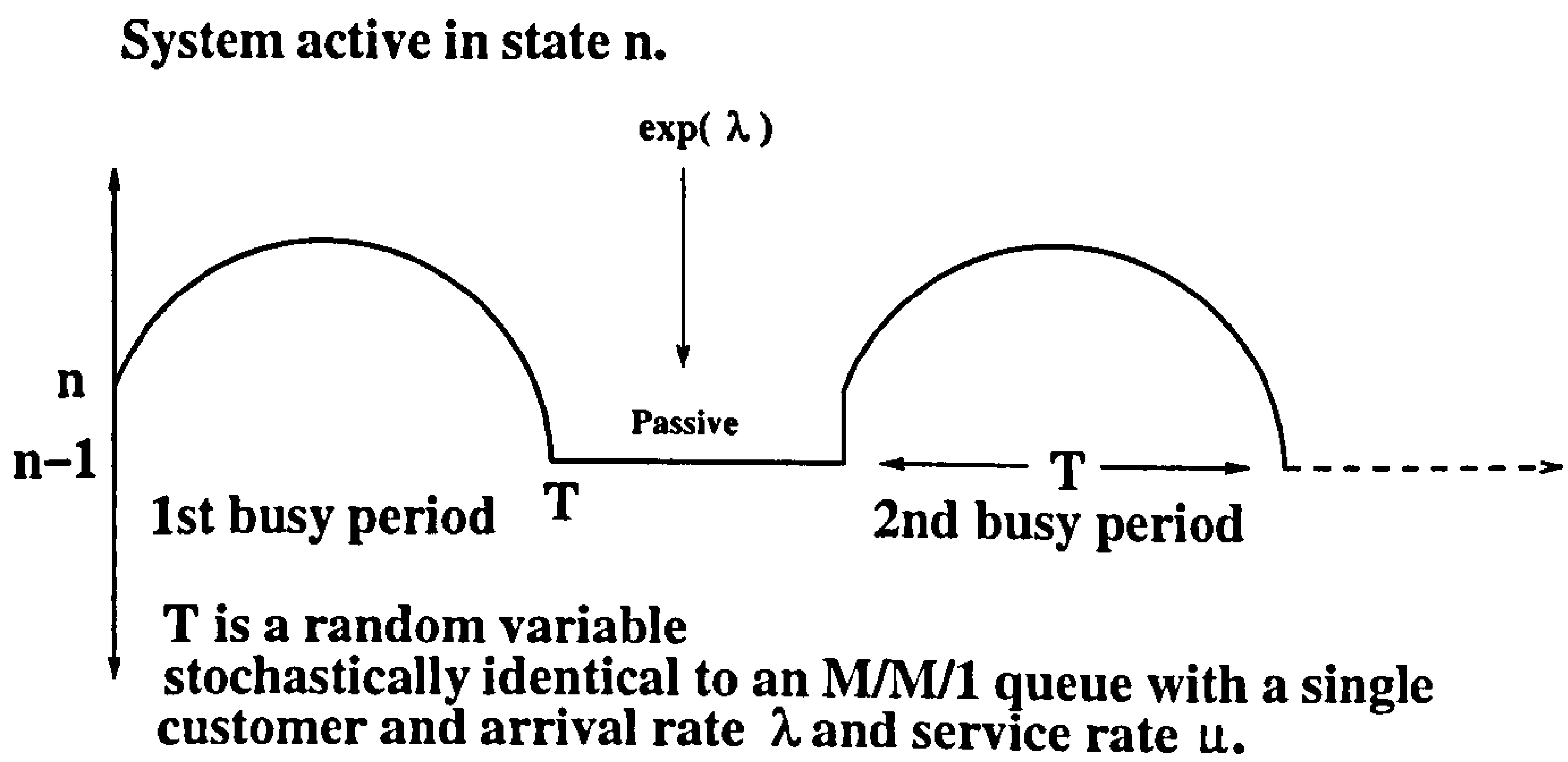

Figure 3.3: The active action in n

Under this policy the total expected discounted costs incurred over an infinite time horizon will be:

$$\left\{\bar{C}(n,\alpha) + E\left(e^{-\alpha T}\right)\{\alpha C(n-1) - \bar{W}_\alpha(n)\}(\alpha+\lambda)^{-1}\right\}\left(1 - \lambda E\left(e^{-\alpha T}\right)(\alpha+\lambda)^{-1}\right)^{-1} \quad (3.15)$$

where the expected cost for the initial busy period is

$$\bar{C}(n,\alpha) = E\left\{ \int_0^T C\{N(t)\}\alpha e^{-\alpha t} dt \mid N(0) = n, \text{active} \right\} \tag{3.16}$$

$$= \frac{\alpha C(n)}{\alpha+\lambda+\mu} + \frac{\lambda}{\lambda+\mu}\left\{ \frac{\lambda+\mu}{\alpha+\lambda+\mu}\bar{C}(n+1,\alpha) + \frac{\lambda+\mu}{\alpha+\lambda+\mu}E\left(e^{-\alpha T}\right)\bar{C}(n,\alpha) \right\} \tag{3.17}$$

and $E\left(e^{-\alpha T}\right)\left\{\alpha C(n-1) - \bar{W}_\alpha(n)\right\}(\alpha+\lambda)^{-1}$ is the expected cost for first the passive period which follows. We can therefore use $\bar{C}(n,\alpha) + E\left(e^{-\alpha T}\right)\left\{\alpha C(n-1) - \bar{W}_\alpha(n)\right\}(\alpha+\lambda)^{-1}$ as the first term in the infinite geometric series illustrated in Figure 3.3 and in which $\lambda E\left(e^{-\alpha T}\right)(\alpha+\lambda)^{-1}$ is the ratio of the series.

We note, that rearranging (3.17) gives;

$$\bar{C}(n,\alpha)\left\{\alpha+\lambda+\mu-\lambda E\left(e^{-\alpha T}\right)\right\} = \alpha C(n) + \lambda\bar{C}(n+1,\alpha) \tag{3.18}$$

which is used in later proofs.

3.3.2 Passive action in n

Now we consider the policy whereby we choose the passive action in n i.e.

(i) active for states $\{n+1, n+2, ...\}$

(ii) passive for states $\{0, ..., n\}$

In Figure 3.4 the system is initially in state n and the passive action is in operation. The active action will only begin after some period of time with distribution $\exp(\lambda)$ after which, the arrival of a customer results in the system entering the state $n+1$. The system then switches to the active action and this continues until the queue returns for the first time to state n. The length of this active period is stochastically identical to the random variable T above in Subsection 3.3.1 (i.e. the busy period for *active in n*)

As with the *active in n* process, once the system returns to n, the process is repeated ad infinitum. Under this policy, the total expected discounted cost to infinity is

$$\left\{\alpha C(n) - \bar{W}_\alpha(n) + \lambda\bar{C}(n+1,\alpha)\right\}(\alpha+\lambda)^{-1}\left\{\left(1-\lambda E\left(e^{-\alpha T}\right)(\alpha+\lambda)^{-1}\right)\right\}^{-1}. \tag{3.19}$$

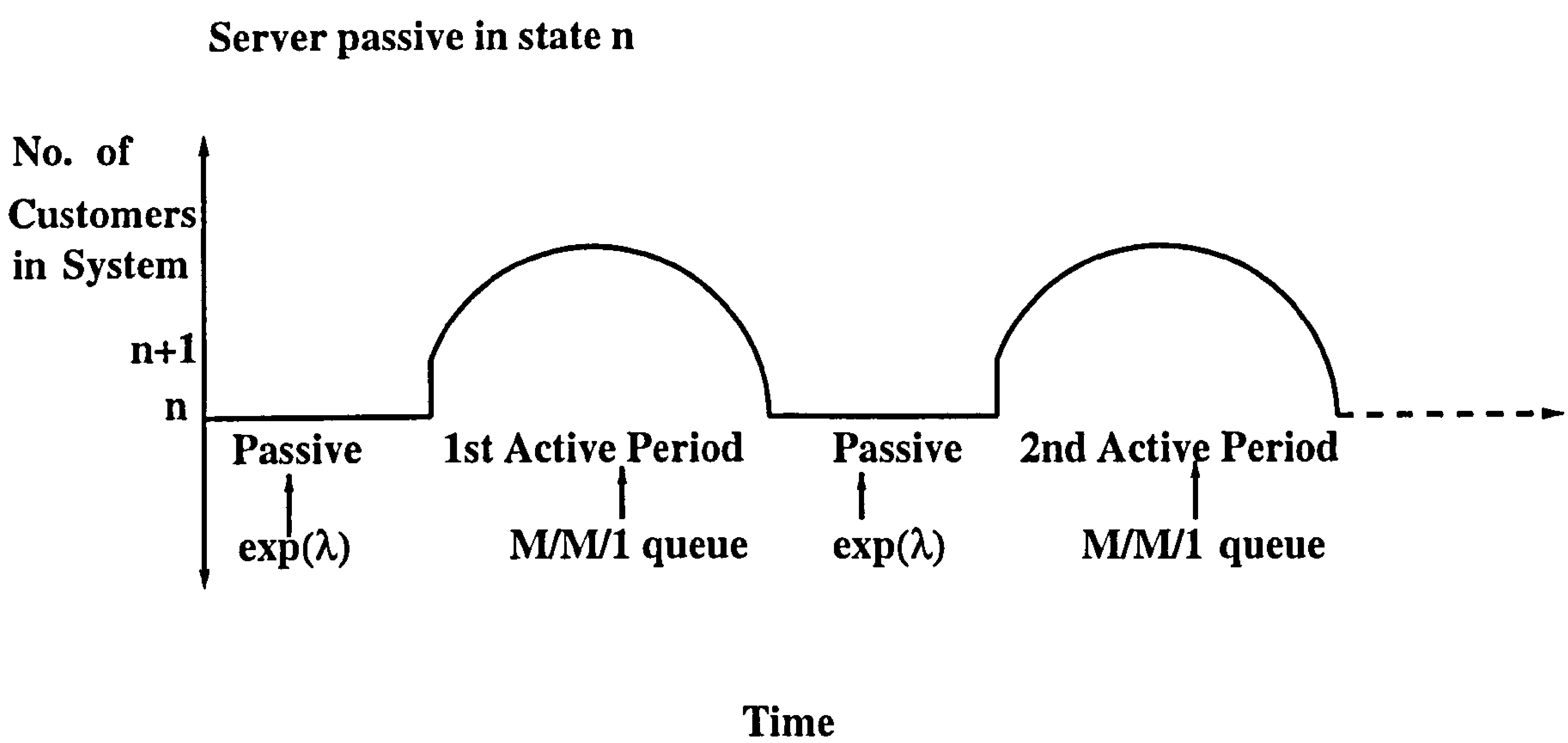

Figure 3.4: The passive action in n

As with (3.15), equation (3.19) may be similarly seen as the sum to infinity of a geometric series. Here the first term is $\left\{\alpha C(n) - \bar{W}_\alpha(n) + \lambda\bar{C}(n+1,\alpha)\right\}(\alpha+\lambda)^{-1}$ and the ratio $\lambda E\left(e^{-\alpha T}\right)(\alpha+\lambda)^{-1}$.

3.4 The Index

Both (3.15) and (3.19) are expressions for the optimal cost of the restless bandit, i.e. the same. Thus, it is possible to equate them in order to obtain an expression for the assumed index, $\bar{W}_\alpha(n)$. This yields from (3.15) and (3.19) that

$$\bar{W}_\alpha(n) = E\left(e^{-\alpha T}\right)\left\{\alpha\bar{C}(n,\alpha)\left\{1-E\left(e^{-\alpha T}\right)\right\}^{-1} - \alpha C(n-1)\right\}\left\{1-E\left(e^{-\alpha T}\right)\right\}^{-1} \quad n \in \mathbb{Z}^+. \tag{3.20}$$

The r.h.s. of (3.20) can be thought of as the discounted rate at which the holding cost rate is reduced by serving the class in state n. In Lemma 3.3, we prove that our proposed index $\bar{W}_\alpha(n)$ is increasing. Here we take $\bar{W}_\alpha(0) = 0$

Lemma 3.3 $\bar{W}_\alpha(n)$ *is increasing in n.*

Proof

From (3.16), we can deduce that;

$$\bar{C}(n,\alpha)\left\{1-E\left(e^{-\alpha T}\right)\right\}^{-1} = \frac{E\left[\int_0^T C\{N(t)\}\alpha e^{-\alpha t}dt \mid N(0)=n\right]}{E\left[\int_0^T \alpha e^{-\alpha t}dt\right]}$$

$$= \sum_{m=0}^{\infty} C(n+m)\,x_m \qquad (3.21)$$

where the set $\{x_m; m \geq 0\}$ form a probability mass function on $\mathbb{N}$. Given an M/M/1 queue, where the arrival and service rates are λ and μ respectively, having a single customer present at time 0 and T the duration of the first busy period, then

$$x_m = \frac{E\left[\int_0^T I_m(s)\,\alpha e^{-\alpha s}ds\right]}{E\left[\int_0^T \alpha e^{-\alpha t}dt\right]}$$

where

$$I_m(s) = \begin{cases} 1, & \text{if } m \text{ customers are present at time } s, \\ 0, & \text{otherwise}, \quad s \in \mathbb{R}^+, \quad m \in \mathbb{N} \end{cases}$$

From (3.21) we can write:

$$\{\bar{C}(n+1,\alpha)-\bar{C}(n,\alpha)\}\left\{1-E\left(e^{-\alpha T}\right)\right\}^{-1} = \sum_{m=0}^{\infty}\{C(n+1+m)-C(n+m)\}x_m$$

$$\geq C(n+1)-C(n) \geq C(n)-C(n-1),$$

$$n \in \mathbb{Z}^+ \qquad (3.22)$$

as C is increasing convex. It, therefore follows from (3.20) and (3.22) that

$$\bar{W}_\alpha(n+1) \geq \bar{W}_\alpha(n), n \in \mathbb{Z}^+,$$

as required. The same method also gives us the result

$$\bar{W}_\alpha(1) \geq 0 = \bar{W}_\alpha(0)$$

and this completes the proof.

Lemma 3.4 *If $\bar{W}_\alpha(m) \leq W < \bar{W}_\alpha(m+1)$ then the policy for the restless bandit in which the server is passive in states $\{0,1,...,m\}$ and active otherwise is optimal, $m \in \mathbb{N}$.*

Proof Fix $W \in [\bar{W}_\alpha(m), \bar{W}_\alpha(m+1))$ and let $\hat{C}(., \alpha, W)$ represent the value function for the policy defined in the Lemma.

We prove the Lemma by showing that $\hat{C}(., \alpha, W)$ satisfies the optimality equations (3.6) - (3.8). It follows from these that we need to show that:

$$\mu\left\{\hat{C}(n,\alpha,W) - \hat{C}(n-1,\alpha,W)\right\} \geq W, n \geq m+1 \tag{3.23}$$

and

$$\mu\left\{\hat{C}(n,\alpha,W) - \hat{C}(n-1,\alpha,W)\right\} \leq W, n \leq m \tag{3.24}$$

In order to prove (3.23) and (3.24), we consider four separate cases, the first of which is given below.

CASE 1: $\mu\left\{\hat{C}(m+1,\alpha,W) - \hat{C}(m,\alpha,W)\right\} \geq W$

We are seeking to show that

$$\mu\left\{\hat{C}(m+1,\alpha,W) - \hat{C}(m,\alpha,W)\right\} \geq W$$

and therefore, we write

$$\mu\left\{\hat{C}(m+1,\alpha,W) - \hat{C}(m,\alpha,W)\right\} < W$$

and seek a contradiction. Note that, from (3.15)

$$\begin{aligned}&\hat{C}(m+1,\alpha,W)\\ &= \left\{\bar{C}(m+1,\alpha) + E\left(e^{-\alpha T}\right)\left\{\alpha C(m) - W\right\}(\alpha+\lambda)^{-1}\right\}\left(1 - \lambda E\left(e^{-\alpha T}\right)(\alpha+\lambda)^{-1}\right)^{-1}\end{aligned}$$

as the server is active in state $m+1$ and from (3.19)

$$\begin{aligned}&\hat{C}(m,\alpha,W) = \\ &\left\{\alpha C(m) - W + \lambda\bar{C}(m+1,\alpha)\right\}(\alpha+\lambda)^{-1}\left\{\left(1 - \lambda E\left(e^{-\alpha T}\right)(\alpha+\lambda)^{-1}\right)\right\}^{-1}\end{aligned} \tag{3.25}$$

as the server is passive in state m.

Thus, we assume that

$$\begin{aligned}&\mu\Big\{\bar{C}(m+1,\alpha) + E\left(e^{-\alpha T}\right)\left\{\alpha C(m) - W\right\}(\alpha+\lambda)^{-1}\\ &-\left\{\alpha C(m) - W + \lambda\bar{C}(m+1,\alpha)\right\}(\alpha+\lambda)^{-1}\Big\}A^{-1} < W,\end{aligned} \tag{3.26}$$

where

$$A = \left\{1 - \lambda E\left(e^{-\alpha T}\right)(\alpha+\lambda)^{-1}\right\} = \left(\alpha+\lambda-\lambda E\left(e^{-\alpha T}\right)\right)(\alpha+\lambda)^{-1} \tag{3.27}$$

and

$$\mu = E\left(e^{-\alpha T}\right)\left(\alpha+\lambda-\lambda E\left(e^{-\alpha T}\right)\right)\left(1-E\left(e^{-\alpha T}\right)\right)^{-1} \textit{ from } (3.14). \tag{3.28}$$

Now

$$\mu A^{-1} = E\left(e^{-\alpha T}\right)(\alpha+\lambda)\left(1-E\left(e^{-\alpha T}\right)\right)^{-1}$$

and, therefore (3.26) becomes

$$\frac{E\left(e^{-\alpha T}\right)}{1-E\left(e^{-\alpha T}\right)}\Big\{\bar{C}(m+1,\alpha)(\alpha+\lambda) + E\left(e^{-\alpha T}\right)\left\{\alpha C(m) - W\right\}$$
$$-\left\{\alpha C(m) - W + \lambda\bar{C}(m+1,\alpha)\right\}\Big\} < W.$$

Rearranging, we obtain

$$\left\{\alpha E\left(e^{-\alpha T}\right)\bar{C}(m+1,\alpha) - \alpha E\left(e^{-\alpha T}\right)C(m)\left(1-E\left(e^{-\alpha T}\right)\right)\right\}\left(1-E\left(e^{-\alpha T}\right)\right)^{-1}$$
$$< W\left(1-2E\left(e^{-\alpha T}\right)+E\left(e^{-\alpha T}\right)^2\right)\left(1-E\left(e^{-\alpha T}\right)\right)^{-1} = \left(1-E\left(e^{-\alpha T}\right)\right)W.$$

Simplifying this gives

$$\left\{\alpha E\left(e^{-\alpha T}\right)\bar{C}(m+1,\alpha)\left(1-E\left(e^{-\alpha T}\right)\right)^{-1} - \alpha E\left(e^{-\alpha T}\right)C(m)\right\}\left(1-E\left(e^{-\alpha T}\right)\right)^{-1} < W.$$

But we have that

$$\left\{\alpha E\left(e^{-\alpha T}\right)\bar{C}(m+1,\alpha)\left(1-E\left(e^{-\alpha T}\right)\right)^{-1} - \alpha E\left(e^{-\alpha T}\right)C(m)\right\}\left(1-E\left(e^{-\alpha T}\right)\right)^{-1}$$
$$= \bar{W}_\alpha(m+1),$$

and hence deduce that

$$\bar{W}_\alpha(m+1) < W.$$

This is a contradiction of our initial assumption that $\bar{W}_\alpha(m) \leq W < \bar{W}_\alpha(m+1)$ and Lemma 3.1. Hence

$$\mu\left\{\hat{C}(m+1,\alpha,W) - \hat{C}(m,\alpha,W)\right\} \geq W$$

as required.

CASE 2:

$$\mu\left\{\hat{C}(n,\alpha,W)-\hat{C}(n-1,\alpha,W)\right\}\geq W,\quad n\geq m+1$$

We prove this by an induction. The initial case holds as this was proved in Case 1 above. We assume that the inequality holds for every value of $m+1\leq n\leq k$ and deduce it for $n=k+1$. Under the given policy, the server will be active for k and $k+1$. Thus, we write:

$$\hat{C}(k+1,\alpha,W)=\bar{C}(k+1,\alpha)+E\left(e^{-\alpha T}\right)\hat{C}(k,\alpha,W)$$

and

$$\hat{C}(k,\alpha,W)=\bar{C}(k,\alpha)+E\left(e^{-\alpha T}\right)\hat{C}(k-1,\alpha,W).$$

We thus seek to prove that

$$\mu\left\{\bar{C}(k+1,\alpha)-\bar{C}(k,\alpha)+E\left(e^{-\alpha T}\right)\left(\hat{C}(k,\alpha,W)-\hat{C}(k-1,\alpha,W)\right)\right\}\geq W. \quad (3.29)$$

Now, by induction

$$\mu\left\{\hat{C}(k,\alpha,W)-\hat{C}(k-1,\alpha,W)\right\}\geq W.$$

It follows that

$$\mu E\left(e^{-\alpha T}\right)\left\{\hat{C}(k,\alpha,W)-\hat{C}(k-1,\alpha,W)\right\}\geq E\left(e^{-\alpha T}\right)W. \quad (3.30)$$

Subtracting (3.30) from (3.29) implies that if we can show that

$$\mu\left\{\bar{C}(k+1,\alpha)-\bar{C}(k,\alpha)\right\}\geq\left(1-E\left(e^{-\alpha T}\right)\right)W \quad (3.31)$$

then Case 2 will be proved.

Substituting from (3.18), the l.h.s. of (3.31) becomes;

$$\mu\left\{\alpha C(k+1)+\lambda\bar{C}(k+2,\alpha)-\alpha C(k)-\lambda\bar{C}(k+1,\alpha)\right\}\left(\alpha+\lambda+\mu-\lambda E\left(e^{-\alpha T}\right)\right)^{-1},$$

which, when substituting for μ from (3.28) is

$$E\left(e^{-\alpha T}\right)\left\{\alpha C(k+1)+\lambda\bar{C}(k+2,\alpha)-\alpha C(k)-\lambda\bar{C}(k+1,\alpha)\right\}.$$

Again from (3.18), we now substitute for $\lambda\bar{C}(k+2,\alpha)$ so we now have to show that

$$E\left(e^{-\alpha T}\right)\left\{\alpha C(k+1)+\bar{C}(k+1,\alpha)\left\{\alpha+\lambda+\mu-\lambda E\left(e^{-\alpha T}\right)\right\}\right.$$
$$\left.-\alpha C(k+1)-\alpha C(k)-\lambda\bar{C}(k+1,\alpha)\right\}\geq\left(1-E\left(e^{-\alpha T}\right)\right)W.$$

Finally, substituting for μ again in the l.h.s. gives us

$$E\left(e^{-\alpha T}\right)\left\{\alpha\bar{C}(k+1,\alpha)\left(1-E\left(e^{-\alpha T}\right)\right)^{-1}-\alpha C(k)\right\}=\left(1-E\left(e^{-\alpha T}\right)\right)\bar{W}_{\alpha}(k+1).$$

Now, by Lemma 3.3 and by the initial hypothesis of Lemma 3.4 we have that

$$\bar{W}_{\alpha}(k+1)\geq\bar{W}_{\alpha}(m+1)>W,$$

and this proves that (3.31) holds and therefore Case 2 is proven as required.

CASE 3: $\mu\left\{\hat{C}(m,\alpha,W)-\hat{C}(m-1,\alpha,W)\right\}\leq W$ (3.32)

Now, as the action is passive for $n\leq m$, we know that

$$\hat{C}(m-1,\alpha,W)=\left\{\alpha C(m-1)-W\right\}(\alpha+\lambda)^{-1}+\lambda\hat{C}(m,\alpha,W)\left\{\alpha+\lambda\right\}^{-1}$$

Therefore, (3.32) can be written as

$$\mu\left\{\left(1-\lambda(\alpha+\lambda)^{-1}\right)\hat{C}(m,\alpha,W)-\left\{\alpha C(m-1)-W\right\}(\alpha+\lambda)^{-1}\right\}\leq W,$$

or

$$\mu\left\{\alpha(\alpha+\lambda)^{-1}\hat{C}(m,\alpha,W)-\left\{\alpha C(m-1)-W\right\}(\alpha+\lambda)^{-1}\right\}\leq W.$$

The action is passive in m so that substituting for $\hat{C}(m,\alpha,W)$ from (3.25) gives us;

$$\mu\left\{\left[\alpha(\alpha+\lambda)^{-1}\left\{\alpha C(m)-W+\lambda\bar{C}(m+1,\alpha)\right\}(\alpha+\lambda)^{-1}\right]A^{-1}\right.$$
$$\left.-\left\{\alpha C(m-1)-W\right\}(\alpha+\lambda)^{-1}\right\}\leq W$$

where A and μ are as in (3.27) and (3.28). Therefore, we have

$$\left\{\alpha E\left(e^{-\alpha T}\right)\left\{\alpha C(m)+\lambda \bar{C}(m+1,\alpha)\right\}\right.$$
$$-\alpha E\left(e^{-\alpha T}\right)\left(\alpha+\lambda-\lambda E\left(e^{-\alpha T}\right)\right)C(m-1)$$
$$\left.+\lambda E\left(e^{-\alpha T}\right)W\left(1-E\left(e^{-\alpha T}\right)\right)\right\}\left(1-E\left(e^{-\alpha T}\right)\right)^{-1}(\alpha+\lambda)^{-1}\leq W$$

and by substituting for $\alpha C(m)+\lambda\bar{C}(m+1,\alpha)$ from (3.18), we have

$$\left\{\alpha E\left(e^{-\alpha T}\right)\left\{\bar{C}(m,\alpha)\left(\alpha+\lambda+\mu-\lambda E\left(e^{-\alpha T}\right)\right)\right\}\right.$$
$$-\alpha E\left(e^{-\alpha T}\right)\left(\alpha+\lambda-\lambda E\left(e^{-\alpha T}\right)\right)C(m-1)$$
$$\left.+\lambda E\left(e^{-\alpha T}\right)W\left(1-E\left(e^{-\alpha T}\right)\right)\right\}\left(1-E\left(e^{-\alpha T}\right)\right)^{-1}(\alpha+\lambda)^{-1}\leq W.$$

Rearranging and removing a factor of $\left(\alpha+\lambda-\lambda E\left(e^{-\alpha T}\right)\right)(\alpha+\lambda)^{-1}$ gives us

$$\alpha E\left(e^{-\alpha T}\right)\bar{C}(m,\alpha)\left(1-E\left(e^{-\alpha T}\right)\right)^{-2}-\alpha E\left(e^{-\alpha T}\right)C(m-1)\left(1-E\left(e^{-\alpha T}\right)\right)^{-1}\leq W$$

or

$$W_\alpha(m)\leq W,$$

which holds by our fixing of the initial value of m. Hence Case 3 holds as required.

CASE 4: $\mu\left\{\hat{C}(n,\alpha,W)-\hat{C}(n-1,\alpha,W)\right\}\leq W,\qquad n\leq m$

We seek to prove this by induction. The initial case, $n=m$, holds from our proof of Case 3 above and we assume the inequality holds for $k+1\leq n\leq m$ and we deduce it for $n=k$.

Now, we know that for $n\leq k$, the passive action is selected. Thus

$$\hat{C}(k,\alpha,W)=(\alpha C(k)-W)(\alpha+\lambda)^{-1}+\lambda(\alpha+\lambda)^{-1}\hat{C}(k+1,\alpha,W)$$

and

$$\hat{C}(k-1,\alpha,W)=(\alpha C(k-1)-W)(\alpha+\lambda)^{-1}+\lambda(\alpha+\lambda)^{-1}\hat{C}(k,\alpha,W).$$

Hence, we seek to show

$$\mu\left\{(\alpha C(k)-\alpha C(k-1))(\alpha+\lambda)^{-1}+\lambda(\alpha+\lambda)^{-1}\left(\hat{C}(k+1,\alpha,W)-\hat{C}(k,\alpha,W)\right)\right\}\leq W. \quad (3.33)$$

We know from the induction that

$$\mu\left\{ \left(\hat{C}(k+1,\alpha,W) - \hat{C}(k,\alpha,W)\right) \right\} \leq W$$

and so we can write

$$\lambda(\alpha+\lambda)^{-1}\mu\left\{ \left(\hat{C}(k+1,\alpha,W) - \hat{C}(k,\alpha,W)\right) \right\} \leq \lambda(\alpha+\lambda)^{-1} W. \tag{3.34}$$

Therefore, subtracting (3.34) from (3.33) it is sufficient to show that

$$\mu\left\{ (\alpha C(k) - \alpha C(k-1))(\alpha+\lambda)^{-1} \right\} \leq \left(1 - \lambda(\alpha+\lambda)^{-1}\right) W.$$

or

$$\mu\left\{ C(k) - C(k-1) \right\} \leq W.$$

Utilising (3.22) and (3.31) we can state;

$$\mu\{C(k) - C(k-1)\} \leq \mu\{\bar{C}(k,\alpha) - \bar{C}(k-1,\alpha)\}\{1 - E\left(e^{-\alpha T}\right)\}^{-1} \tag{3.35}$$

$$= \bar{W}_{\alpha}(k). \tag{3.36}$$

Thus, by Lemma 3.3, we can write

$$\bar{W}_{\alpha}(k) \leq \bar{W}_{\alpha}(m) \leq W. \tag{3.37}$$

The induction is proven and Case 4 holds. We have thus established (3.23) and (3.24) and hence proved Lemma 3.4. We now go on to prove our first theorem.

Theorem 3.1 (Indexability and the Whittle index for discounted costs) . *The restless bandit is indexable with Whittle index* $W_{\alpha}(n) = \bar{W}_{\alpha}(n), n \in \mathbb{N}$.

Proof By Lemma 3.4 , we can write:

$$\Pi_{\alpha}(W) = \{0, 1, ..., n\}, \qquad \bar{W}_{\alpha}(n) \leq W < \bar{W}_{\alpha}(n+1), n \in \mathbb{N}. \tag{3.38}$$

We have shown indexability by Lemma 3.4 and, from (3.38) and Definition 3.2, we have shown $\bar{W}_{\alpha}(n)$ to be the Whittle index in state n.

As indicated in Lemma 3.1, we can now go on to seek a Whittle index, $W : \mathbb{N} \to \mathbb{R}^+$ for the average cost problem by finding the limit

$$W(n) = \lim_{\alpha \to 0} W_\alpha(n)$$
$$= \lim_{\alpha \to 0} \bar{W}_\alpha(n), \quad n \in \mathbb{N}. \tag{3.39}$$

by Theorem 3.1. From (3.20) and (3.39), we obtain the following result.

Theorem 3.2 (The Whittle index for average costs.) *The Whittle index for the average cost problem is given by* $W(0) = 0$ *and*

$$W(n) = \left[\bar{C}(n)\{E(T)\}^{-1} - C(n-1)\right]\{E(T)\}^{-1}, \quad n \in \mathbb{Z}^+, \tag{3.40}$$
$$= \frac{\mu(\mu - \lambda)}{\lambda}\left[E\{C(n-1+N)\} - C(n-1)\right], \quad n \in \mathbb{Z}^+ \tag{3.41}$$

where in (3.40), *we have*

$$\bar{C}(n) = E\left[\int_0^T C\{N(t)\}dt \mid N(0) = n\right], n \in \mathbb{Z}^+, \tag{3.42}$$

and N, *in* (3.41), *is a random variable with probability mass function*

$$P(N = n) = \rho^n(1 - \rho), \quad n \in \mathbb{N}, \tag{3.43}$$

where $\rho = \frac{\lambda}{\mu}$.

Proof Utilising (3.20) and (3.16) we can write;

$$\lim_{\alpha \to 0} \bar{W}_\alpha(n) = \lim_{\alpha \to 0}\left[E\left(e^{-\alpha T}\right)\alpha E\left\{\int_0^T C\{N(t)\}\alpha e^{-\alpha t}dt \mid N(0) = n\right\}\left\{1 - E\left(e^{-\alpha T}\right)\right\}^{-2}\right.$$
$$\left. -\alpha C(n-1)\left\{1 - E\left(e^{-\alpha T}\right)\right\}^{-1}\right] \quad n \in \mathbb{Z}^+. \tag{3.44}$$

$$= \lim_{\alpha \to 0}\left\{\frac{\alpha^2 E\left(e^{-\alpha T}\right)E\left[\int_0^T C\{N(t)\}e^{-\alpha t}dt \mid N(0) = n\right]}{\left(1 - E\left(e^{-\alpha T}\right)\right)^2} - \frac{\alpha E\left(e^{-\alpha T}\right)C(n-1)}{\left(1 - E\left(e^{-\alpha T}\right)\right)}\right\}$$

Utilising the fact that

$$E\left(e^{-\alpha T}\right) = E(1 - \alpha T) + O\left(\alpha^2\right)$$

we have

$$= \lim_{\alpha \to 0} \frac{\alpha^2 E\left(e^{-\alpha T}\right) E\left[\int_0^T C\{N(t)\} e^{-\alpha t} dt \mid N(0) = n\right]}{\alpha^2 (E(T))^2} - \frac{\alpha E\left(e^{-\alpha T}\right) C(n-1)}{\alpha E(T)}$$

Letting $\alpha \to 0$ we obtain that,

$$W(n) = \frac{E\left[\int_0^T C\{N(t)\} dt \mid N(0) = n\right]}{(E(T))^2} - \frac{(C(n-1)}{E(T)},$$

as required. Thus, we have derived (3.40). For (3.41), we note from standard renewal theory arguments, that the average cost incurred by adopting the passive action in states $\{0, 1, \ldots, n-1\}$ and the active action otherwise, when $C(n)$ is the cost rate in state $n \in \mathbb{N}$ is given by

$$\left\{\bar{C}(n) + C(n-1)\lambda^{-1}\right\}\left\{E(T) + \lambda^{-1}\right\}^{-1} = E\{C(n-1+N)\} \qquad (3.45)$$

where N is a random variable with the steady state distribution for the number of customers present in an M/M/1 system with arrival rate λ and service rate μ, as in (3.43). (3.41) follows from (3.40) and (3.45) and the substitution of $E(T) = (\mu - \lambda)^{-1}$. $W(0) = 0$ is immediate from (3.39)

3.5 PCL-Indexability

We note here that Niño-Mora (2001*b*) offers an alternative demonstration of Whittle-indexability for restless bandits and index derivation. He uses the notion of partial conservation laws to determine indexability (hence the term; PCL-indexability). The notion of PCL has been developed from the generalised conservation laws described in Chapter 2. Here, we offer a summary of the main ideas propounded.

Assume we wish to schedule a stochastic system serving a countably infinite number of customer classes indexed by the natural numbers, $\mathbb{N}$. Let U denote the collection of admissible scheduling policies. We wish, say, to minimise some linear objective

$$\sum_{i \in \mathbb{N}} c_i x_i^u$$

where $c_i > 0$ is a cost rate for customer class i and x_i^u is some performance measure for class i under scheduling policy u. Niño-Mora (2001*b*) shows that, when a number of partial work conservation laws are satisfied by the system, the minimisation problem is solved by an index policy for *some* choices of the cost rate vector $\mathbf{c}$. To determine whether or not a particular choice is admissible, an adaptive greedy algorithm is applied. A system satisfying PCL and with a cost rate vector in the admissible class is defined as PCL-indexable.

In a further publication, Niño-Mora (2001*a*), uses the above ideas to develop sufficient conditions for the Whittle-indexability of countable state restless bandits in terms of model parameters. He then goes on to show that the restless bandit model associated with a multi-class M/M/1 system satisfies these conditions and is therefore PCL-indexable.

Using the PCL approach, he is able to develop a closed form expression for the discounted index by using a modified version of the adaptive greedy algorithm. He goes on to obtain an average cost index by seeking a limit as α tends to infinity. The analysis is complex but PCL-indexability has the advantage of offering an alternative approach to analysis when simple direct arguments, as used in our work, may not be possible.

Chapter 4

An Evaluation of a Whittle index policy in two simple cases with average costs.

In the previous chapter, we followed the prescription of Whittle (1988) for the development of an index appropriate to our multi-class queueing system. Much of the chapter was devoted to the demonstration that the system was indeed indexable. We then, by means of a simple argument, devised the form of the index for the discounted costs version of the problem and then devised the index for the undiscounted problem by taking the limit as $\alpha \rightarrow 0$.

In this chapter, we assess the performance of the K-class average cost Whittle index policy, derived for the stochastic problem of the previous chapter, for both a two class system and a three class queueing system. The results of numerical investigations into the performance of the index policy in some simple cases involving two and three customer classes and with quadratic costs are presented. In the two customer class cases, the index policies clearly outperform the threshold policies proposed by Ansell et al. (1999).

We first consider the form that the index takes when the cost function is quadratic in n.

4.1 The form of the index for a quadratic cost function

We consider a cost function of the form

$$C(n) = bn + cn^2. \tag{4.1}$$

In Chapter 3, we had the following equation for the Whittle index for the average cost problem: $W(0) = 0$ and

$$W(n) = \frac{\mu(\mu-\lambda)}{\lambda}\left[E\{C(n-1+N)\} - C(n-1)\right], \quad n \in \mathbb{Z}^+.$$

For the cost function in (4.1), this may be written in the following form

$$W(n) = \left\{\sum_{m=0}^{\infty} b(n+m)x_m + \sum_{m=0}^{\infty} c(n+m)^2 x_m - b(n-1) - c(n-1)^2\right\}(\mu-\lambda).$$

Further, we know that, in our case $x_m = (1-\rho)\rho^{m-1}$ and, therefore we can rewrite the r.h.s. of the above as

$$\left\{\left[\sum_{m=1}^{\infty} b(n+m-1)(1-\rho)\rho^{m-1} + \sum_{m=1}^{\infty} c(n+m-1)^2(1-\rho)\rho^{m-1}\right] \right.$$
$$\left. -b(n-1) - c(n-1)^2\right\}(\mu-\lambda).$$

Thus

$$\begin{aligned}W(n) &= \left\{b(n-1) + \sum_{m=1}^{\infty} bm(1-\rho)\rho^{m-1} + c\left(n^2-2n+1\right)\right. \\ &\quad + \sum_{m=1}^{\infty} 2c(n-1)m(1-\rho)\rho^{m-1} + \sum_{m=1}^{\infty} cm^2(1-\rho)\rho^{m-1} \\ &\quad \left. -b(n-1) - c\left(n^2-2n+1\right)\right\}(\mu-\lambda) \\ &= \left\{\sum_{m=1}^{\infty} bm(1-\rho)\rho^{m-1} + \sum_{m=1}^{\infty} 2c(n-1)m(1-\rho)\rho^{m-1}\right. \\ &\quad \left. + \sum_{m=1}^{\infty} cm^2(1-\rho)\rho^{m-1}\right\}(\mu-\lambda).\end{aligned}$$

Now multiplying through by $\frac{\varrho}{\rho}$, we can rewrite the above as

$$= \left\{ \sum_{m=0}^{\infty} bm\,(1-\rho)\,\rho^m + \sum_{m=0}^{\infty} 2c\,(n-1)\,m\,(1-\rho)\,\rho^m + \sum_{m=0}^{\infty} cm^2\,(1-\rho)\,\rho^m \right\} \left(\frac{\mu}{\lambda}\right)(\mu-\lambda)$$

$$= \left\{ \{b+2c\,(n-1)\} \sum_{m=0}^{\infty} mp\,(m) + c \sum_{m=0}^{\infty} m^2 p\,(m) \right\} \left(\frac{\mu}{\lambda}\right)(\mu-\lambda)$$

$$= \left\{ \{b+2c\,(n-1)\} \frac{\lambda}{(\mu-\lambda)} + c\frac{\lambda\,(\mu+\lambda)}{(\mu-\lambda)^2} \right\} \left(\frac{\mu}{\lambda}\right)(\mu-\lambda)$$

which gives the final form of the index as

$$\begin{aligned} W\,(n) &= \{b+2c\,(n-1)\}\,\mu + c\frac{(\mu+\lambda)\,\mu}{(\mu-\lambda)} \\ &= b\mu + \frac{c\,(3\lambda-\mu)\,\mu}{\mu-\lambda} + 2c\mu n,\ n \in \mathbb{Z}^+,\ k = 1,2,..,K. \end{aligned} \tag{4.2}$$

We note that this index becomes $b\mu$ when $c = 0$ and thus is optimal when the costs are linear. Note also that from calculations similar to the above we can infer that if the cost rate $C\,(n)$ is a polynomial in n of order p, then the index derived, $W\,(n)$, will be a polynomial of order $p-1$. In the quadratic case, the index is linear in the queue length.

Thus, the Whittle index policy is one which allocates service to whichever class, of those in the system, has the highest Whittle index, as given by

$$W_k\,(n_k) = b_k\mu_k + \frac{c_k\,(3\lambda_k-\mu_k)\,\mu_k}{\mu_k-\lambda_k} + 2c_k\mu_k n_k,\ n_k \in \mathbb{Z}^+,\ k = 1,2,..,K. \tag{4.3}$$

We first assess the performance of the Whittle index policy for a two class system with quadratic costs; $C_k\,(n) = b_k n + c_k n^2$ where $k = 1,2$.

Thus, from (3.4) we are seeking an admissible control to minimise average costs, given by

$$C^{OPT} = min_{u\in U}\tilde{E}_u \left\{ \sum_{k=1}^{K} b_1N_1 + b_2N_2 + c_1N_1^2 + c_2N_2^2 \right\}. \tag{4.4}$$

In order to assess the performance of the Whittle index policy, we will consider a range of policies for this problem, namely the threshold policies and linear switching policies. We compare these with one another and against exact values of, and lower bounds on, the minimised achievable cost, C^{OPT}.

4.2 Whittle index policy for a two customer class problem

In a two customer class problem, a Whittle index policy will select a customer for service between the two customer classes, when both have jobs present in the system, on the basis of the class indices of (4.3) where $k = 1, 2$.

The index policy dictates that at each decision epoch whichever class is non empty and has the larger value of

$$W_1(n_1) = b_1\mu_1 + \frac{c_1(3\lambda_1 - \mu_1)\mu_1}{(\mu_1 - \lambda_1)} + 2c_1\mu_1 n_1 \tag{4.5}$$

and

$$W_2(n_2) = b_2\mu_2 + \frac{c_2(3\lambda_2 - \mu_2)\mu_2}{(\mu_2 - \lambda_2)} + 2c_2\mu_2 n_2 \tag{4.6}$$

is chosen for service. If $W_1\{N_1(t)\} \geq W_2\{N_2(t)\}$, then a class 1 customer is served at time t, assuming $N_1(t) > 0$. Otherwise a class 2 customer is served, assuming $N_2(t) > 0$. Such a policy clearly belongs to the class based on linear switching curves. We assess the performance of the Whittle index policy for a two class M/M/1 system with quadratic costs: $C_k(n) = b_k n + c_k n^2$, $k = 1, 2$ where we seek an admissible control policy to minimise C^u. As usual, we write $C^{OPT} = \inf_{u \in U} C^u$, where

$$C^u = \tilde{E}_u\left\{b_1 N_1 + b_2 N_2 + C_1 N_1^2 + c_2 N_2^2\right\}. \tag{4.7}$$

We compare the performance of Whittle index policies in a number of different systems by considering their associated costs against the minimised achievable cost, C^{OPT} and a semidefinite lower bound for it.

4.2.1 A specific linear switching curve

In the two customer case, we are able to use the index to derive a policy based on a *specific* linear switching curve. We are then able to compute the cost under such a policy.

It is straightforward to show that under the index policy in (4.5) and (4.6), priority is given to type 1 customers until queue length of type 2 reaches the line $n_2 = \alpha n_1 + \beta$, where

$$\alpha = \frac{c_1\mu_1}{c_2\mu_2},$$
$$\beta = \frac{(\beta_1 - \beta_2)}{2c_2\mu_2},$$

with

$$\beta_k = (b_k\mu_k - 2c_k\mu_k) + \frac{c_k\mu_k(\lambda_k + \mu_k)}{(\mu_k - \lambda_k)}, \quad k = 1, 2. \tag{4.8}$$

Thus it was possible to use the methods of Chapter 2 to perform a numerical study to investigate the performance of index policies. As in Chapter 2 we calculate best costs achievable under threshold and linear switching policies (found by searching α and β space) and calculate a semidefinite lower bound.

We also calculate a value for C^{OPT} using the value-iteration algorithm (see Tijms (1994)). In general the calculation of C^{OPT} quickly becomes unviable as the dimensionality of the problems increases. We use it here because the two customer problems concerned are simple enough to allow us to do so.

4.3 Calculation of C^{OPT} via the value-iteration algorithm

The minimum cost, C^{OPT}, incurred when an optimal policy is operated on the two class, and later three class problems, was computed for the numerical study by the dynamic programming method of value-iteration. See Tijms (1994). The value-iteration algorithm calculates recursively a series of value functions which approximate the minimal average cost per unit of time. The value functions give us lower and upper bounds on the minimal average cost rate and, in our particular Markov decision problem, they approximate this rate to a chosen degree of accuracy. The algorithm is given in Table 4.1. The notation is that of Tijms (1994).

The value-iteration algorithm computes the value functions $V_n(i)$ for $n = 1, 2, ...$ recursively from

$$V_n(i) = min_{a \in A(i)} \left\{ c_i(a) + \sum_{j \in I} p_{ij}(a) V_{n-1}(j) \right\}, \quad i \in I.$$

$V_0(i)$, $i \in I$ is arbitrarily chosen and $V_n(i)$ is the minimal total expected costs when there are n time periods remaining. The current state is i, I is the state space, $c_i(a)$ is the cost of taking action a, from set of possible actions, $A(i)$ in state i and a terminal cost of $V_0(j)$ is incurred when the system ends in state j. For the two customer problem, this

Table 4.1: The value-iteration algorithm.

Step 0. Arbitrarily choose $V_0(i)$ such that $0 \leq V_0(i) \leq min_a c_i(a)$ for all $i \in I$. Let $n := 1$

Step 1. Calculate the value function $V_n(i)$, $i \in I$, from

$$V_n(i) = min_{a \in A(i)} \left\{ c_i(a) + \sum_{j \in I} p_{ij}(a) V_{n-1}(j) \right\} \tag{4.9}$$

and find $R(n)$ as a stationary policy whose actions minimise the r.h.s. of (4.9) for all $i \in I$.

Step 2. Calculate the bounds

$$m_n = min_{j \in I} \{V_n(j) - V_{n-1}(j)\} \quad \text{and} \quad M_n = max_{j \in I} \{V_n(j) - V_{n-1}(j)\}$$

The algorithm stops with $R(n)$ as a stationary policy when $0 \leq M_n - m_n \leq \epsilon m_n$, where ϵ is the required degree of accuracy. Otherwise go to step 3.

Step 3. $n := n + 1$ and go to step 1.

is equivalent to the following recursion (where we retain our notation rather than that of Tijms):

$$\begin{aligned}
C_{t+1}(n_1,n_2) &= \min\left\{\frac{b_1n_1+b_2n_2+c_1n_1^2+c_2n_2^2}{\lambda_1+\lambda_2+\mu_1}\right. \\
&\quad +\frac{\lambda_1C_t(n_1+1,n_2)+\lambda_2C_t(n_1,n_2+1)+\mu_1C_t(n_1-1,n_2)}{\lambda_1+\lambda_2+\mu_1}; \\
&\quad \frac{b_1n_1+b_2n_2+c_1n_1^2+c_2n_2^2}{\lambda_1+\lambda_2+\mu_2} \\
&\quad \left.+\frac{\lambda_1C_t(n_1+1,n_2)+\lambda_2C_t(n_1,n_2+1)+\mu_2C_t(n_1,n_2-1)}{\lambda_1+\lambda_2+\mu_2}\right\} \\
&\qquad (n_1,n_2)\in\left(\mathbb{Z}^+\right)^2
\end{aligned}$$

$$C_{t+1}(n_1,0) = \frac{b_1n_1+c_1n_1^2+\lambda_1C_t(n_1+1,0)+\lambda_2C_t(n_1,1)+\mu_1C_t(n_1-1,0)}{(\lambda_1+\lambda_2+\mu_1)}, \quad n_1\in\mathbb{Z}^+$$

$$C_{t+1}(0,n_2) = \frac{b_2n_2+c_2n_2^2+\lambda_1C_t(1,n_2)+\lambda_2C_t(0,n_2+1)+\mu_2C_t(0,n_2-1)}{(\lambda_1+\lambda_2+\mu_2)}, \quad n_2\in\mathbb{Z}^+$$

$$C_{t+1}(0,0) = \frac{\lambda_1C_t(1,0)+\lambda_2C_t(0,1)}{(\lambda_1+\lambda_2)}, \quad t\in\mathbb{N}.$$

Our calculations were computed over a state space large enough to give a result with the required degree of accuracy, $\epsilon = 0.000000001$. The exact size of a state space used in a given calculation tended to be a trade-off between computation time and achieving the required degree of accuracy. It is likely that this method would not be a realistic possibility for larger problems. With this in mind, we also produced lower bounds for C^{OPT} based on the achievable region approach, a more computationally efficient method, as described and utilised in Chapter 2.

4.4 Two class problem; results

The numerical results presented in Tables 4.2 and 4.3 are for problems with $\lambda_1 = 1$, $\lambda_2 = 5$, $\mu_1 = 3$ and $\mu_2 = 12$.

C^T represents the costs associated with the best threshold policy. C^{SW} represents the costs associated with best linear switching policy found by a search strategy. C^{IND} represents the costs associated with the index policy. C^{OPT} is the best achievable cost, calculated via dynamic programming. C^{SD} is a cost from deriving a semidefinite lower bound on C^{OPT}. We shall describe in due course how to obtain C^{SD}

By definition we have

$$C^T \geq C^{SW} \geq C^{OPT} \geq C^{SD}$$

$$C^{IND} \geq C^{SW} \geq C^{OPT} \geq C^{SD}$$

In Table 4.2, the cost coefficients are ; $b_1 = 5$, $b_2 = 1$, while, in Table 4.3, $b_1 = 4$, $b_2 = 2$ and c_1 and c_2 are as indicated in both tables.

It is clear that the performance of the Whittle index policy is, in every case, close to optimal. The search strategy has sometimes resulted in a marginally lower cost (closer to C^{OPT}) being found but the Whittle index policy has the advantage of being easy to calculate and avoids lengthy and computationally expensive search procedures. We also note, as an aside to the main thrust of our work, that C^{SD} provides a bound on C^{OPT} sufficiently tight to support the use of the achievable region approach in larger problems where the computation of C^{OPT} is not viable.

Table 4.2: Two customer type problems.

c_1	c_2	C^T	C^{SW}	C^{IND}	C^{OPT}	C^{SD}
0.1	0.1	9.344	9.334	9.335	9.334	9.305
0.1	0.2	9.581	9.575	9.575	9.575	9.566
0.1	0.5	10.101	10.101	10.101	10.101	10.095
0.1	1.0	10.969	10.969	10.969	10.969	10.964
0.1	2.0	12.703	12.703	12.703	12.703	12.700
0.2	0.1	9.926	9.882	9.885	9.882	9.858
0.2	0.2	10.244	10.199	10.199	10.199	10.184
0.2	0.5	10.764	10.763	10.763	10.763	10.740
0.2	1.0	11.631	11.631	11.631	11.631	11.619
0.2	2.0	13.366	13.366	13.366	13.366	13.358
0.5	0.1	11.476	11.275	11.276	11.273	11.242
0.5	0.2	12.053	11.906	11.917	11.906	11.866
0.5	0.5	12.752	12.700	12.701	12.699	12.604
0.5	1.0	13.620	13.615	13.615	13.615	13.544
0.5	2.0	15.354	15.354	15.354	15.354	15.316
1.0	0.1	13.513	13.016	13.026	13.014	12.898
1.0	0.2	14.742	14.304	14.307	14.303	14.205
1.0	0.5	15.962	15.713	15.725	15.707	15.494
1.0	1.0	16.933	16.848	16.848	16.848	16.638
1.0	2.0	18.668	18.660	18.660	18.660	18.517
2.0	0.1	16.473	15.404	15.427	15.390	15.089
2.0	0.2	19.074	17.990	17.990	17.982	17.778
2.0	0.5	21.799	20.992	21.096	20.992	20.660
2.0	1.0	23.482	22.917	22.917	22.917	22.418
2.0	2.0	25.294	25.146	25.146	25.146	24.703

Table 4.3: Two customer type problems contd.

c_1	c_2	C^T	C^{SW}	C^{IND}	C^{OPT}	C^{SD}
0.1	0.1	8.550	8.549	8.550	8.550	8.520
0.1	0.2	8.724	8.723	8.724	8.724	8.709
0.1	0.5	9.244	9.244	9.244	9.244	9.238
0.1	1.0	10.112	10.111	10.112	10.112	10.109
0.1	2.0	11.846	11.846	11.846	11.846	11.845
0.2	0.1	9.213	9.209	9.213	9.213	9.100
0.2	0.2	9.387	9.385	9.386	9.386	9.327
0.2	0.5	9.907	9.906	9.907	9.907	9.883
0.2	1.0	10.774	10.772	10.774	10.774	10.762
0.2	2.0	12.509	12.508	12.509	12.509	12.503
0.5	0.1	11.201	11.131	11.133	11.131	10.688
0.5	0.2	11.375	11.346	11.346	11.345	11.020
0.5	0.5	11.895	11.889	11.890	11.890	11.747
0.5	1.0	12.762	12.756	12.762	12.762	12.687
0.5	2.0	14.497	14.491	14.497	14.497	14.459
1.0	0.1	14.515	13.813	13.813	13.809	12.945
1.0	0.2	14.688	14.321	14.329	14.319	13.662
1.0	0.5	15.208	15.100	15.100	15.100	14.637
1.0	1.0	16.076	16.052	16.052	16.051	15.780
1.0	2.0	17.810	17.796	17.808	17.808	17.660
2.0	0.1	19.314	17.525	17.525	17.512	16.619
2.0	0.2	20.592	19.042	19.042	19.025	18.175
2.0	0.5	21.776	20.896	20.896	20.896	19.974
2.0	1.0	22.703	22.351	22.351	22.351	21.560
2.0	2.0	24.437	24.359	24.359	24.356	23.846

4.5 Whittle index policy for the three class problem

We now extend our work to assess the performance of the Whittle index policy in the more complicated three class system with quadratic costs. We now have

$$C_k(n) = b_k n + c_k n^2 \text{ where } k = 1, 2, 3.$$

Thus, we are seeking an admissible control to minimise average costs

$$C^{OPT} = min_{u \in U} \tilde{E}_u \left\{ b_1 N_1 + b_2 N_2 + b_3 N_3 + c_1 N_1^2 + c_2 N_2^2 + c_3 N_3^2 \right\}.$$

In order to assess the performance of the Whittle index policy, we again consider a range of policies for this problem. These include the threshold policies and linear switching policies as defined in Chapter 2. We compare these with one another and against exact values of and lower bounds on the minimised achievable cost, C^{OPT}.

It follows from (4.3) that a Whittle index policy will select a customer for service from the three customer classes, when all have jobs present in the system, on the basis of the class indices

$$W_k(n_k) = b_k \mu_k + \frac{c_k (3\lambda_k - \mu_k) \mu_k}{\mu_k - \lambda_k} + 2c_k \mu_k n_k, \; n_k \in \mathbb{Z}^+, \; k = 1, 2, 3.$$

If $n_1, n_2, n_3 > 0$ then the Whittle index policy dictates that the customer class with the highest value of $W_k(n_k)$ will be given priority. Clearly, if at time t one of the customer classes has no customers present in the system, then service is allocated to whichever of the remaining two classes with jobs in the system has the larger index value. Finally, if, at time t, there is only one non-zero customer class present in the system, then the server is allocated to that class. In order to calculate the expected cost incurred under the index policy, we need to obtain the steady state distribution of the system under the policy. We will apply the methods of Chapter 2 to carry out the three-class analysis.

The joint steady state distribution for the system under the index policy

$$p_{i,j,k} = \lim_{t \to \infty} P\{N_1(t) = i, N_2(t) = j, N_3(t) = k\}$$

satisfies the set of balance equations given below.

$$\Big\{\lambda_1 + \lambda_2 + \lambda_3 + \mu_1 L_1 + \mu_2 L_2 + \mu_3 L_3 + \mu_1 L_4 + \mu_2 L_5 + \mu_3 L_6$$

$$+\mu_1 L_7 + \mu_2 L_8 + \mu_3 L_9 + \mu_1 L_{10} + \mu_2 L_{11} + \mu_3 L_{12}\Big\} p_{i,j,k}$$

$$= \lambda_1 p_{i-1,j,k} + \lambda_2 p_{i,j-1,k} + \lambda_3 p_{i,j,k-1}$$

$$+\mu_1 L_{13} p_{i+1,j,k} + \mu_2 L_{14} p_{i,j+1,k} + \mu_3 L_{15} p_{i,j,k+1}$$

$$+\mu_1 L_{16} p_{i+1,j,k} + \mu_2 L_{17} p_{i,j+1,k} + \mu_3 L_{18} p_{i,j,k+1}$$

$$+\mu_1 L_{19} p_{i+1,j,k} + \mu_2 L_{20} p_{i,j+1,k} + \mu_3 L_{21} p_{i,j,k+1}$$

$$+\mu_1 L_{22} p_{i+1,j,k} + \mu_2 L_{23} p_{i,j+1,k} + \mu_3 L_{24} p_{i,j,k+1}$$

where

$L_1 = \delta\,(i{>}0, j{>}0, k{=}0, \alpha_1 i + \beta_1 \geq \alpha_2 j + \beta_2)$

$L_2 = \delta\,(i{>}0, j{>}0, k{=}0, \alpha_2 j + \beta_2 > \alpha_1 i + \beta_1)$

$L_3 = \delta\,(i{>}0, j{=}0, k{>}0, \alpha_3 k + \beta_3 > \alpha_1 i + \beta_1)$

$L_4 = \delta\,(i{>}0, j{=}0, k{>}0, \alpha_1 i + \beta_1 \geq \alpha_3 k + \beta_3)$

$L_5 = \delta\,(i{=}0, j{>}0, k{>}0, \alpha_2 j + \beta_2 \geq \alpha_3 k + \beta_3)$

$L_6 = \delta\,(i{=}0, j{>}0, k{>}0, \alpha_3 k + \beta_3 > \alpha_2 j + \beta_2)$

$L_7 = \delta\,(i{>}0, j{=}0, k{=}0)$

$L_8 = \delta\,(i{=}0, j{>}0, k{=}0)$

$L_9 = \delta\,(i{=}0, j{=}0, k{>}0)$

$L_{10} = \delta\,(i{>}0, j{>}0, k{>}0, \alpha_1 i + \beta_1 \geq \alpha_2 j + \beta_2, \alpha_1 i + \beta_1 \geq \alpha_3 k + \beta_3)$

$L_{11} = \delta\,(i{>}0, j{>}0, k{>}0, \alpha_2 j + \beta_2 > \alpha_1 i + \beta_1, \alpha_2 j + \beta_2 \geq \alpha_3 k + \beta_3)$

$L_{12} = \delta\,(i{>}0, j{>}0, k{>}0, \alpha_3 k + \beta_3 > \alpha_1 i + \beta_1, \alpha_3 k + \beta_3 > \alpha_2 j + \beta_2)$

$L_{13} = \delta\,(i+1{>}0, j{>}0, k{=}0, \alpha_1 i + 1 + \beta_1 \geq \alpha_2 j + \beta_2)$

$L_{14} = \delta\,(i{>}0, j+1{>}0, k{=}0, \alpha_2 j + 1 + \beta_2 > \alpha_1 i + \beta_1)$

$L_{15} = \delta\,(i{>}0, j{=}0, k+1{>}0, \alpha_3 k + 1 + \beta_3 > \alpha_1 i + \beta_1)$

$L_{16} = \delta\,(i+1{>}0, j{=}0, k{>}0, \alpha_1 i + 1 + \beta_1 \geq \alpha_3 k + \beta_3)$

$$L_{17} = \delta\left(i{=}0, j+1{>}0, k{>}0, \alpha_2 j + 1 + \beta_2 {\geq} \alpha_3 k + \beta_3\right)$$
$$L_{18} = \delta\left(i{=}0, j{>}0, k+1{>}0, \alpha_3 k + 1 + \beta_3 {>} \alpha_2 j + \beta_2\right)$$
$$L_{19} = \delta\left(i+1{>}0, j{=}0, k{=}0\right)$$
$$L_{20} = \delta\left(i{=}0, j+1{>}0, k{=}0\right)$$
$$L_{21} = \delta\left(i{=}0, j{=}0, k+1{>}0\right)$$
$$L_{22} = \delta\left(i+1{>}0, j{>}0, k{>}0, \alpha_1 i + 1 + \beta_1 {\geq} \alpha_2 j + \beta_2, \alpha_1 i + 1 + \beta_1 {\geq} \alpha_3 k + \beta_3\right)$$
$$L_{23} = \delta\left(i{>}0, j+1{>}0, k{>}0, \alpha_2 j + 1 + \beta_2 {>} \alpha_1 i + \beta_1, \alpha_2 j + 1 + \beta_2 {\geq} \alpha_3 k + \beta_3\right)$$
$$L_{24} = \delta\left(i{>}0, j{>}0, k+1{>}0, \alpha_3 k + 1 + \beta_3 {>} \alpha_1 i + \beta_1, \alpha_3 k + 1 + \beta_3 {>} \alpha_2 j + \beta_2\right)$$

and

$$p_{-1,j,k} = p_{i,-1,k} = p_{i,j,-1} = 0. \tag{4.10}$$

From (4.2)

$$\alpha_i = 2c_i\mu_i$$

and

$$\beta_i = b_i\mu_i + \frac{c_i\left(3\lambda_i - \mu_i\right)\mu_i}{\left(\mu_i - \lambda_i\right)}, \quad i = 1, 2, 3.$$

We use δ as an indicator function where $\delta(B) = 1$ if B is true and 0 otherwise. Also, we employ the convention whereby if two classes have the same index, the policy chooses the one with lower numerical identifier.

In the numerical study of the three class problem, we obtain solutions of these balance equations by again applying the power series algorithm and epsilon algorithm as we did in Chapter 2. First, we introduce a conformal mapping for the balance equations so that we write $\frac{\lambda_i\theta}{1+G-G\theta}$ for the λ_i as follows

$$\left\{\left(\lambda_1+\lambda_2+\lambda_3\right)\frac{\theta}{1+G-G\theta} + \mu_1 L_1 + \mu_2 L_2 + \mu_3 L_3 + \mu_1 L_4 + \mu_2 L_5 + \mu_3 L_6 \right.$$
$$\left. + \mu_1 L_7 + \mu_2 L_8 + \mu_3 L_9 + \mu_1 L_{10} + \mu_2 L_{11} + \mu_3 L_{12}\right\} p_{i,j,k} =$$
$$\left(\lambda_1 p_{i-1,j,k} + \lambda_2 p_{i,j-1,k} + \lambda_3 p_{i,j,k-1}\right)\frac{\theta}{1+G-G\theta} + \mu_1 L_{13} p_{i+1,j,k} + \mu_2 L_{14} p_{i,j+1,k}$$
$$+ \mu_3 L_{15} p_{i,j,k+1} + \mu_1 L_{16} p_{i+1,j,k} + \mu_2 L_{17} p_{i,j+1,k} + \mu_3 L_{18} p_{i,j,k+1} + \mu_1 L_{19} p_{i+1,j,k}$$
$$+ \mu_2 L_{20} p_{i,j+1,k} + \mu_3 L_{21} p_{i,j,k+1} + \mu_1 L_{22} p_{i+1,j,k} + \mu_2 L_{23} p_{i,j+1,k} + \mu_3 L_{24} p_{i,j,k+1}.$$

We now replace $p_{i,j,k}$ by $\theta^{i+j+k}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k}$ to obtain

$$
\begin{aligned}
&\Big\{ (\lambda_1+\lambda_2+\lambda_3)\frac{\theta}{1+G-G\theta}+\mu_1L_1+\mu_2L_2+\mu_3L_3+\mu_1L_4+\mu_2L_5 \\
&+\mu_3L_6+\mu_1L_7+\mu_2L_8+\mu_3L_9+\mu_1L_{10}+\mu_2L_{11}+\mu_3L_{12}\Big\} \\
&\times\theta^{i+j+k}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k} \;=\; \Big\{\lambda_1\theta^{i-1+j+k}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i-1,j,k}+\lambda_2\theta^{i+j-1+k}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j-1,k} \\
&\quad+\lambda_3\theta^{i+j+k-1}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k-1}\Big\}\frac{\theta}{1+G-G\theta} \\
&\quad+\mu_1L_{13}\theta^{i+1+j+k}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i+1,j,k}+\mu_2L_{14}\theta^{i+j+1+k}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j+1,k} \\
&\quad+\mu_3L_{15}\theta^{i+j+k+1}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k+1}+\mu_1L_{16}\theta^{i+1+j+k}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i+1,j,k} \\
&\quad+\mu_2L_{17}\theta^{i+j+1+k}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j+1,k}+\mu_3L_{18}\theta^{i+j+k+1}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k+1} \\
&\quad+\mu_1L_{19}\theta^{i+1+j+k}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i+1,j,k}+\mu_2L_{20}\theta^{i+j+1+k}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j+1,k} \\
&\quad+\mu_3L_{21}\theta^{i+j+k+1}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k+1}+\mu_1L_{22}\theta^{i+1+j+k}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i+1,j,k} \\
&\quad+\mu_2L_{23}\theta^{i+j+1+k}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j+1,k}+\mu_3L_{24}\theta^{i+j+k+1}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k+1}.
\end{aligned}
$$

Multiplying by $(1+G-G\theta)$ gives us

$$
\begin{aligned}
&(\lambda_1+\lambda_2+\lambda_3)\,\theta^{i+j+k+1}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k+1}\\
&+\mu_1\{L_1+L_4+L_7+L_{10}\}\theta^{i+j+k}\,(1+G-G\theta)\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k}\\
&+\mu_2\{L_2+L_5+L_8+L_{11}\}\theta^{i+j+k}\,(1+G-G\theta)\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k}\\
&+\mu_3\{L_3+L_6+L_9+L_{12}\}\theta^{i+j+k}\,(1+G-G\theta)\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k}=\\
&\quad\lambda_1\theta^{i+j+k}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i-1,j,k}+\lambda_2\theta^{i+j+k}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j-1,k}+\lambda_3\theta^{i+j+k}\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k-1}\\
&\quad+\mu_1\{L_{13}+L_{16}+L_{19}+L_{22}\}\theta^{i+1+j+k}\,(1+G-G\theta)\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i+1,j,k}\\
&\quad+\mu_2\{L_{14}+L_{17}+L_{20}+L_{23}\}\theta^{i+j+1+k}\,(1+G-G\theta)\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j+1,k}\\
&\quad+\mu_3\{L_{15}+L_{18}+L_{21}+L_{24}\}\theta^{i+j+k+1}\,(1+G-G\theta)\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k+1}.
\end{aligned}
$$

Taking out a factor of θ^{i+j+k} gives

$$
\begin{aligned}
&(\lambda_1+\lambda_2+\lambda_3)\,\theta\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k+1}+\mu_1\{L_1+L_4+L_7+L_{10}\}\,(1+G-G\theta)\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k}\\
&+\mu_2\{L_2+L_5+L_8+L_{11}\}\,(1+G-G\theta)\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k}\\
&+\mu_3\{L_3+L_6+L_9+L_{12}\}\,(1+G-G\theta)\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k}=\\
&\qquad\lambda_1\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i-1,j,k}+\lambda_2\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j-1,k}+\lambda_3\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k-1}\\
&\qquad+\mu_1\{L_{13}+L_{16}+L_{19}+L_{22}\}\theta\,(1+G-G\theta)\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i+1,j,k}\\
&\qquad+\mu_2\{L_{14}+L_{17}+L_{20}+L_{23}\}\theta\,(1+G-G\theta)\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j+1,k}\\
&\qquad+\mu_3\{L_{15}+L_{18}+L_{21}+L_{24}\}\theta\,(1+G-G\theta)\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k+1},
\end{aligned}
$$

and on expansion, we have

$$
\begin{aligned}
&(\lambda_1+\lambda_2+\lambda_3)\,\theta\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k+1}+\mu_1\{L_1+L_4+L_7+L_{10}\}\,(1+G)\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k}\\
&+\mu_2\{L_2+L_5+L_8+L_{11}\}\,(1+G)\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k}\\
&+\mu_3\{L_3+L_6+L_9+L_{12}\}\,(1+G)\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k}\\
&-\mu_1\{L_1+L_4+L_7+L_{10}\}G\theta\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k}-\mu_2\{L_2+L_5+L_8+L_{11}\}G\theta\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k}\\
&-\mu_3\{L_3+L_6+L_9+L_{12}\}G\theta\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k}=\\
&\qquad\lambda_1\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i-1,j,k}+\lambda_2\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j-1,k}+\lambda_3\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k-1}\\
&\qquad+\mu_1\{L_{13}+L_{16}+L_{19}+L_{22}\}\,(1+G)\,\theta\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i+1,j,k}\\
&\qquad+\mu_2\{L_{14}+L_{17}+L_{20}+L_{23}\}\,(1+G)\,\theta\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j+1,k}\\
&\qquad+\mu_3\{L_{15}+L_{18}+L_{21}+L_{24}\}\,(1+G)\,\theta\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k+1}\\
&\qquad-\mu_1\{L_{13}+L_{16}+L_{19}+L_{22}\}G\theta^2\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i+1,j,k}\\
&\qquad-\mu_2\{L_{14}+L_{17}+L_{20}+L_{23}\}G\theta^2\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j+1,k}\\
&\qquad-\mu_3\{L_{15}+L_{18}+L_{21}+L_{24}\}G\theta^2\sum_{h=0}^{\infty}\theta^h\hat{p}_{h,i,j,k+1}.
\end{aligned}
$$

We equate coefficients to obtain

$$(\lambda_1+\lambda_2+\lambda_3)\,\hat{p}_{h-1,i,j,k+1}+(1+G)\Big\{\mu_1\big(L_1+L_4+L_7+L_{10}\big)$$
$$+\mu_2\big(L_2+L_5+L_8+L_{11}\big)+\mu_3\big(L_3+L_6+L_9+L_{12}\big)\Big\}\hat{p}_{h,i,j,k}$$
$$-G\Big\{\mu_1\big(L_1+L_4+L_7+L_{10}\big)+\mu_2\big(L_2+L_5+L_8+L_{11}\big)$$
$$+\mu_3\big(L_3+L_6+L_9+L_{12}\big)\Big\}\hat{p}_{h-1,i,j,k} \quad =$$
$$\lambda_1\hat{p}_{h,i-1,j,k}+\lambda_2\hat{p}_{h,i,j-1,k}+\lambda_3\hat{p}_{h,i,j,k-1}$$
$$+(1+G)\Big\{\mu_1\big(L_{13}+L_{16}+L_{19}+L_{22}\big)\hat{p}_{h-1,i+1,j,k}+\mu_2\big(L_{14}+L_{17}+L_{20}+L_{23}\big)\hat{p}_{h-1,i,j+1,k}$$
$$+\mu_3\big(L_{15}+L_{18}+L_{21}+L_{24}\big)\hat{p}_{h-1,i,j,k+1}\Big\}-G\Big\{\mu_1\big(L_{13}+L_{16}+L_{19}+L_{22}\big)\hat{p}_{h-2,i+1,j,k}$$
$$+\mu_2\big(L_{14}+L_{17}+L_{20}+L_{23}\big)\hat{p}_{h-2,i,j+1,k}+\mu_3\big(L_{15}+L_{18}+L_{21}+L_{24}\big)\hat{p}_{h-2,i,j,k+1}\Big\}.$$

Finally, making $\hat{p}_{h,i,j,k}$ the subject of the equation gives us

$$\begin{aligned}
\hat{p}_{h,i,j,k} \;=\; & \lambda_1\hat{p}_{h,i-1,j,k}+\lambda_2\hat{p}_{h,i,j-1,k}+\lambda_3\hat{p}_{h,i,j,k-1} \\
& -(\lambda_1+\lambda_2+\lambda_3)\,\hat{p}_{h-1,i,j,k+1}+G\Big\{\mu_1\big(L_1+L_4+L_7+L_{10}\big) \\
& +\mu_2\big(L_2+L_5+L_8+L_{11}\big)+\mu_3\big(L_3+L_6+L_9+L_{12}\big)\Big\}\hat{p}_{h-1,i,j,k} \\
& +(1+G)\Big\{\mu_1\big(L_{13}+L_{16}+L_{19}+L_{22}\big)\hat{p}_{h-1,i+1,j,k} \\
& +\mu_2\big(L_{14}+L_{17}+L_{20}+L_{23}\big)\hat{p}_{h-1,i,j+1,k} \\
& +\mu_3\big(L_{15}+L_{18}+L_{21}+L_{24}\big)\hat{p}_{h-1,i,j,k+1}\Big\} \\
& -G\Big\{\mu_1\big(L_{13}+L_{16}+L_{19}+L_{22}\big)\hat{p}_{h-2,i+1,j,k} \\
& +\mu_2\big(L_{14}+L_{17}+L_{20}+L_{23}\big)\hat{p}_{h-2,i,j+1,k} \\
& +\mu_3\big(L_{15}+L_{18}+L_{21}+L_{24}\big)\hat{p}_{h-2,i,j,k+1}\Big\} \\
& \Big\{(1+G)\Big\{\mu_1\big(L_1+L_4+L_7+L_{10}\big)+\mu_2\big(L_2+L_5+L_8+L_{11}\big) \\
& +\mu_3\big(L_3+L_6+L_9+L_{12}\big)\Big\}\Big\}^{-1}.
\end{aligned}$$

The solution of these recursions enables the computation of the $p_{i,j,k}$.

4.6 Approximation of the Achievable Region in a three class system.

We once again, as in Chapter 2, use the potential function method to formulate a set of constraints which will yield a relaxation of the achievable region. Under uniformisation, we have that $\sum_{r=1}^{3}[\lambda_r + \mu_r] = 1$ and τ_k is the sequence of transition times in the uniformised Markov chain. As in Chapter 2, $B_r(t)$ denotes the event that the server is busy with a class r customer at time t, where $r = 1, 2, 3$. Similarly, $\overline{B}_r(t)$ denotes the event that the server is *not* busy with a class r customer at time t.

We are characterising the set of possible first and second moments of the three queue lengths i.e. $\{E(N_1), E(N_2), E(N_3), E(N_1^2), E(N_2^2), E(N_3^2),\}$. This is the achievable region. We are concerned with the random behaviour of the potential function $R(t)$ under a general control policy for the three class system. Their derivation is given below and the potential function is given in (4.11), namely:

$$R(t) = f(1)N_1(t) + f(2)N_2(t) + f(3)N_3(t) \tag{4.11}$$

We first use the recursion

$$\begin{aligned}
E\left[R^2(\tau_{k+1})\,|\mathrm{N}(\tau_k)\right] &= \sum_{r=1}^{3}\lambda_r\left(R(\tau_k)+f(r)\right)^2 + \sum_{r=1}^{3}\mu_r\delta\left(B_r\{\tau_k\}\right)\left[\left(R(\tau_k)-f(r)\right)^2\right] \\
&\quad + \sum_{r=1}^{3}\mu_r\delta\left(\overline{B}_r\{\tau_k\}\right)R^2(\tau_k) \\
&= \sum_{r=1}^{3}\lambda_r\left(R^2(\tau_k)+2f(r)R(\tau_k)+f^2(r)\right) \\
&\quad + \sum_{r=1}^{3}\mu_r\delta\left(B_r\{\tau_k\}\right)\left[R^2(\tau_k)-2f(r)R(\tau_k)+f^2(r)\right] \\
&\quad + \sum_{r=1}^{3}\mu_r\delta\left(\overline{B}_r\{\tau_k\}\right)R^2(\tau_k),
\end{aligned}$$

and taking expectations gives us

$$\begin{aligned} E\left[E\left[R^2(\tau_{k+1})\,|\mathrm{N}(\tau_k)\right]\right] &= \sum_{r=1}^{3}\lambda_r\left\{E\left[R^2(\tau_k)\right]+2f(r)\,E\left[R(\tau_k)\right]+E\left[f^2(r)\right]\right\} \\ &+\sum_{r=1}^{3}\mu_r\left\{E\left[\delta\left(B_r\{\tau_k\}\right)R^2(\tau_k)\right]-2f(r)\,E\left[\delta\left(B_r\{\tau_k\}\right)R(\tau_k)\right]\right. \\ &\left.+E\left[\delta\left(B_r\{\tau_k\}\right)f^2(r)\right]\right\}+\sum_{r=1}^{3}\mu_r E\left[\delta\left(\overline{B}_r\{\tau_k\}\right)R^2(\tau_k)\right]. \end{aligned}$$

Now, we use the identity (because the system is in steady state)

$$E\left\{E\left[R^2(\tau_{k+1})\,|\mathrm{N}(\tau_k)\right]\right\}=E\left[R^2(\tau_{k+1})\right]=E\left[R^2(\tau_k)\right]. \tag{4.12}$$

Therefore, we can write

$$\begin{aligned} E\left[R^2(\tau_k)\right] &= \sum_{r=1}^{3}\lambda_r E\left[R^2(\tau_k)\right]+\sum_{r=1}^{3}\mu_r E\left[\delta\left(B_r\{\tau_k\}\right)R^2(\tau_k)\right] \\ &+\sum_{r=1}^{3}\mu_r E\left[\delta\left(\overline{B}_r\{\tau_k\}\right)R^2(\tau_k)\right]+\sum_{r=1}^{3}\lambda_r\left\{2f(r)\,E\left[R(\tau_k)\right]+E\left[f^2(r)\right]\right\} \\ &-2\sum_{r=1}^{3}\mu_r f(r)\,E\left[\delta\left(B_r\{\tau_k\}\right)R(\tau_k)\right]+\sum_{r=1}^{3}\mu_r E\left[\delta\left(B_r\{\tau_k\}\right)f^2(r)\right] \\ &=\sum_{r=1}^{3}\lambda_r E\left[R^2(\tau_k)\right]+\sum_{r=1}^{3}\mu_r E\left[R^2(\tau_k)\right]+\sum_{r=1}^{3}\lambda_r\left\{2f(r)\,E\left[R(\tau_k)\right]+E\left[f^2(r)\right]\right\} \\ &-2\sum_{r=1}^{3}\mu_r f(r)\,E\left[\delta\left(B_r\{\tau_k\}\right)R(\tau_k)\right]+\sum_{r=1}^{3}\mu_r E\left[\delta\left(B_r\{\tau_k\}\right)f^2(r)\right] \\ &=\sum_{r=1}^{3}\left[\lambda_r+\mu_r\right]E\left[R^2(\tau_k)\right]+\sum_{r=1}^{3}\lambda_r\left\{2f(r)\,E\left[R(\tau_k)\right]+E\left[f^2(r)\right]\right\} \\ &-2\sum_{r=1}^{3}\mu_r f(r)\,E\left[\delta\left(B_r\{\tau_k\}\right)R(\tau_k)\right]+\sum_{r=1}^{3}\mu_r E\left[\delta\left(B_r\{\tau_k\}\right)f^2(r)\right], \end{aligned}$$

Now because of uniformisation, we can rearrange to obtain

$$\begin{aligned} 0 &= \sum_{r=1}^{3}\lambda_r\left\{2f(r)\,E\left[R(\tau_k)\right]+E\left[f^2(r)\right]\right\}-2\sum_{r=1}^{3}\mu_r f(r)\,E\left[\delta\left(B_r\{\tau_k\}\right)R(\tau_k)\right] \\ &+\sum_{r=1}^{3}\mu_r E\left[\delta\left(B_r\{\tau_k\}\right)f^2(r)\right]. \end{aligned}$$

Now we have that

$$E\left[\delta\left(B_r\left\{\tau_k\right\}\right)\right] = P\left(\text{Server is busy with a type } r \text{ customer}\right) = \frac{\lambda_r}{\mu_r}. \tag{4.13}$$

Thus, we have

$$\begin{aligned} 0 =& 2\sum_{r=1}^{3}\lambda_r f(r) E\left[R(\tau_k)\right] + \sum_{r=1}^{3}\lambda_r E\left[f^2(r)\right] \\ & - 2\sum_{r=1}^{3}\lambda_r\mu_r f(r) E\left[\delta\left(B_r\left\{\tau_k\right\}\right) R(\tau_k)\right] + \sum_{r=1}^{3}\lambda_r E\left[f^2(r)\right] \\ =& \sum_{r=1}^{3}\lambda_r f(r) E\left[R(\tau_k)\right] - \sum_{r=1}^{3}\lambda_r\mu_r f(r) E\left[\delta\left(B_r\left\{\tau_k\right\}\right) R(\tau_k)\right] \\ & + \sum_{r=1}^{3}\lambda_r E\left[f^2(r)\right]. \end{aligned}$$

On expansion we have

$$\begin{aligned} &\lambda_1 f(1) E\left[R(\tau_k)\right] + \lambda_2 f(2) E\left[R(\tau_k)\right] + \lambda_3 f(3) E\left[R(\tau_k)\right] - \mu_1 f(1) E\left[\delta\left(B_1\left\{\tau_k\right\}\right)\right] \\ &-\mu_2 f(2) E\left[\delta\left(B_2\left\{\tau_k\right\}\right)\right] - \mu_3 f(3) E\left[\delta\left(B_3\left\{\tau_k\right\}\right)\right] + \lambda_1 f^2(1) + \lambda_2 f^2(2) + \lambda_3 f^2(3) = 0 \end{aligned}$$

and substituting from (4.11), we have

$$\begin{aligned} &\lambda_1 f(1) E\left[f(1) N_1(\tau_k) + f(2) N_2(\tau_k) + f(3) N_3(\tau_k)\right] \\ &+\lambda_2 f(2) E\left[f(1) N_1(\tau_k) + f(2) N_2(\tau_k) + f(3) N_3(\tau_k)\right] \\ &+\lambda_3 f(3) E\left[f(1) N_1(\tau_k) + f(2) N_2(\tau_k) + f(3) N_3(\tau_k)\right] \\ &-\mu_1 f(1) E\left[\delta\left(B_1\left\{\tau_k\right\}\right)\left\{f(1) N_1(\tau_k) + f(2) N_2(\tau_k) + f(3) N_3(\tau_k)\right\}\right] \\ &-\mu_2 f(2) E\left[\delta\left(B_2\left\{\tau_k\right\}\right)\left\{f(1) N_1(\tau_k) + f(2) N_2(\tau_k) + f(3) N_3(\tau_k)\right\}\right] \\ &-\mu_3 f(3) E\left[\delta\left(B_3\left\{\tau_k\right\}\right)\left\{f(1) N_1(\tau_k) + f(2) N_2(\tau_k) + f(3) N_3(\tau_k)\right\}\right] \\ &+\lambda_1 f^2(1) + \lambda_2 f^2(2) + \lambda_3 f^2(3) = 0. \end{aligned}$$

Rearranging, we obtain

$$\lambda_1 f^2(1)\,E[N_1(\tau_k)] + \lambda_1 f(1)\,f(2)\,E[N_2(\tau_k)] + \lambda_1 f(1)\,f(3)\,E[N_3(\tau_k)]$$
$$+\lambda_2 f(1)\,f(2)\,E[N_1(\tau_k)] + \lambda_2 f^2(2)\,E[N_2(\tau_k)] + \lambda_2 f(2)\,f(3)\,E[N_3(\tau_k)]$$
$$+\lambda_3 f(1)\,f(3)\,E[N_1(\tau_k)] + \lambda_3 f(2)\,f(3)\,E[N_2(\tau_k)] + \lambda_3 f^2(3)\,E[N_3(\tau_k)]$$
$$-\mu_1 f^2(1)\,E[\delta(B_1\{\tau_k\})\,N_1\{\tau_k\}] - \mu_1 f(1)\,f(2)\,E[\delta(B_1\{\tau_k\})\,N_2\{\tau_k\}]$$
$$-\mu_1 f(1)\,f(3)\,E[\delta(B_1\{\tau_k\})\,N_3\{\tau_k\}] - \mu_2 f(1)\,f(2)\,E[\delta(B_2\{\tau_k\})\,N_1\{\tau_k\}]$$
$$-\mu_2 f^2(2)\,E[\delta(B_2\{\tau_k\})\,N_2\{\tau_k\}] - \mu_2 f(2)\,f(3)\,E[\delta(B_2\{\tau_k\})\,N_3\{\tau_k\}]$$
$$-\mu_3 f(1)\,f(3)\,E[\delta(B_3\{\tau_k\})\,N_1\{\tau_k\}] - \mu_3 f(2)\,f(3)\,E[\delta(B_3\{\tau_k\})\,N_2\{\tau_k\}]$$
$$-\mu_3 f^2(3)\,E[\delta(B_3\{\tau_k\})\,N_3\{\tau_k\}] + \lambda_1 f^2(1) + \lambda_2 f^2(2) + \lambda_3 f^2(3) = 0.$$

Taking expectations of the system in steady state and using the substitutions

$$n_r = E[N_r(\tau_k)] \tag{4.14}$$
$$I_{rs} = E[\delta(B_r\{\tau_k\})\,N_s(\tau_k)] \text{ where } r, s, = 1, 2, 3 \tag{4.15}$$

gives us the following sets of equations

$$\lambda_1 f^2(1)\,n_1 - \mu_1 f^2(1)\,I_{11} + \lambda_1 f^2(1) + \lambda_2 f^2(2)\,n_2 - \mu_2 f^2(2)\,I_{22} + \lambda_2 f^2(2)$$
$$+\lambda_3 f^2(3)\,n_3 - \mu_3 f^2(3)\,I_{33} + \lambda_3 f^2(3)$$
$$+\lambda_1 f(1)\,f(2)\,n_2 + \lambda_2 f(1)\,f(2)\,n_1 - \mu_1 f(1)\,f(2)\,I_{12} - \mu_2 f(1)\,f(2)\,I_{21}$$
$$+\lambda_1 f(1)\,f(3)\,n_3 + \lambda_3 f(1)\,f(3)\,n_1 - \mu_1 f(1)\,f(3)\,I_{13} - \mu_3 f(1)\,f(3)\,I_{31}$$
$$+\lambda_2 f(2)\,f(3)\,n_3 + \lambda_3 f(2)\,f(3)\,n_2 - \mu_2 f(2)\,f(3)\,I_{23} - \mu_3 f(2)\,f(3)\,I_{32} = 0.$$

Equating the coefficients of $f^2(1)$, $f(1)f(2)$ etc. we have

$$\lambda_1 n_1 - \mu_1 I_{11} + \lambda_1 = 0 \tag{4.16}$$

$$\lambda_2 n_2 - \mu_2 I_{22} + \lambda_2 = 0 \tag{4.17}$$

$$\lambda_3 n_3 - \mu_3 I_{33} + \lambda_3 = 0 \tag{4.18}$$

$$\lambda_1 n_2 + \lambda_2 n_1 - \mu_1 I_{12} - \mu_2 I_{21} = 0 \tag{4.19}$$

$$\lambda_1 n_3 + \lambda_3 n_1 - \mu_1 I_{13} - \mu_3 I_{31} = 0 \tag{4.20}$$

$$\lambda_2 n_3 + \lambda_3 n_2 - \mu_2 I_{23} - \mu_3 I_{32} = 0. \tag{4.21}$$

The three customer cubic recursion will now be developed. We first note that, from (4.11)

$$\begin{aligned} R^2(t) &= f^2(1) N_1^2(t) + f^2(2) N_2^2(t) + f^2(3) N_3^2(t) + 2f(1) f(2) N_1(t) N_2(t) \\ &\quad +2f(1) f(3) N_1(t) N_3(t) + 2f(2) f(3) N_2(t) N_3(t). \end{aligned}$$

We use the recursion

$$\begin{aligned} E\left[R^3(\tau_{k+1}) | N(\tau_k)\right] &= \sum_{r=1}^{3} \Big\{ \lambda_r (R(\tau_k) + f(r))^3 + \mu_r \delta(B_r\{\tau_k\}) \left[(R(\tau_k) - f(r))^3\right] \\ &\quad + \mu_r \delta\left(\overline{B}_r\{\tau_k\}\right) R^3(\tau_k) \Big\} \\ &= \sum_{r=1}^{3} \Big\{ \lambda_r \left(R^3(\tau_k) + 3f(r) R^2(\tau_k) + 3f^2(r) R(\tau_k) + f^3(r)\right) \\ &\quad + \mu_r \delta(B_r\{\tau_k\}) \left[R^3(\tau_k) - 3f(r) R^2(\tau_k) + 3f^2(r) R(\tau_k) - f^3(r)\right] \\ &\quad + \mu_r \delta\left(\overline{B}_r\{\tau_k\}\right) R^3(\tau_k) \Big\}. \end{aligned}$$

Taking expectations on both sides, as in the case of the quadratic, gives us

$$\begin{aligned} E\left[E\left[R^3(\tau_{k+1}) | N(\tau_k)\right]\right] &= \sum_{r=1}^{3} E\Big\{ \lambda_r \left(R^3(\tau_k) + 3f(r) R^2(\tau_k) + 3f^2(r) R(\tau_k) + f^3(r)\right) \\ &\quad + \mu_r \delta(B_r\{\tau_k\}) \left[R^3(\tau_k) - 3f(r) R^2(\tau_k) + 3f^2(r) R(\tau_k) - f^3(r)\right] \\ &\quad + \mu_r \delta\left(\overline{B}_r\{\tau_k\}\right) R^3(\tau_k) \Big\}. \end{aligned}$$

Now, we use the identity

$$E\Big\{ E\left[R^3(\tau_{k+1}) | N(\tau_k)\right] \Big\} = E\left[R^3(\tau_{k+1})\right] = E\left[R^3(\tau_k)\right]$$

and write

$$E\left[R^3(\tau_k)\right] = \sum_{r=1}^{3}\left\{\lambda_r E\left[R^3(\tau_k)\right] + \mu_r E\left[\delta\left(B_r\{\tau_k\}\right)R^3(\tau_k)\right] + \mu_r E\left[\delta\left(\overline{B}_r\{\tau_k\}\right)R^3(\tau_k)\right]\right\}$$
$$+\sum_{r=1}^{3} E\left\{\lambda_r\left(3f(r)R^2(\tau_k) + 3f^2(r)R(\tau_k) + f^3(r)\right)\right.$$
$$\left.+\mu_r\delta\left(B_r\{\tau_k\}\right)\left(-3f(r)R^2(\tau_k) + 3f^2(r)R(\tau_k) - f^3(r)\right)\right\}$$

$$=\sum_{r=1}^{3}\left\{\lambda_r E\left[R^3(\tau_k)\right] + \mu_r E\left[R^3(\tau_k)\right]\right\}$$
$$+\sum_{r=1}^{3} E\left\{\lambda_r\left(3f(r)R^2(\tau_k) + 3f^2(r)R(\tau_k) + f^3(r)\right)\right.$$
$$\left.+\mu_r\delta\left(B_r\{\tau_k\}\right)\left(-3f(r)R^2(\tau_k) + 3f^2(r)R(\tau_k) - f^3(r)\right)\right\}.$$

This may be rewritten

$$E\left[R^3(\tau_k)\right] = \sum_{r=1}^{3}\left[\lambda_r + \mu_r\right]E\left[R^3(\tau_k)\right] + \sum_{r=1}^{3} E\left\{\lambda_r\left(3f(r)R^2(\tau_k) + 3f^2(r)R(\tau_k) + f^3(r)\right)\right.$$
$$\left.+\mu_r\delta\left(B_r\{\tau_k\}\right)\left(-3f(r)R^2(\tau_k) + 3f^2(r)R(\tau_k) - f^3(r)\right)\right\}$$

and, as $\sum_{r=1}^{m}\left[\lambda_r + \mu_r\right] = 1$

$$E\left[R^3(\tau_k)\right] = E\left[R^3(\tau_k)\right] + \sum_{r=1}^{3} E\left\{\lambda_r\left(3f(r)R^2(\tau_k) + 3f^2(r)R(\tau_k) + f^3(r)\right)\right.$$
$$\left.+\mu_r\delta\left(B_r\{\tau_k\}\right)\left(-3f(r)R^2(\tau_k) + 3f^2(r)R(\tau_k) - f^3(r)\right)\right\}.$$

Therefore, we can equate the remaining terms of the equation to zero to obtain

$$0 = \sum_{r=1}^{3} E\left\{\lambda_r\left(3f(r)R^2(\tau_k) + 3f^2(r)R(\tau_k) + f^3(r)\right)\right.$$
$$\left.+\mu_r\delta\left(B_r\{\tau_k\}\right)\left(-3f(r)R^2(\tau_k) + 3f^2(r)R(\tau_k) - f^3(r)\right)\right\}.$$

Now, from (4.13), we have

$$\sum_{r=1}^{3}\mu_r E\left[\delta\left(B_r\{\tau_k\}\right)\right]f^3(r) = \sum_{r=1}^{3}\mu_r\frac{\lambda_r}{\mu_r}f^3(r) = \sum_{r=1}^{3}\lambda_r f^3(r),$$

and thus

$$0=\sum_{r=1}^{3} E\Big\{3\lambda_r f(r) R^2(\tau_k) + 3\lambda_r f^2(r) R(\tau_k) + \lambda_r f^3(r) - 3\mu_r \delta(B_r\{\tau_k\}) f(r) R^2(\tau_k)$$

$$+3\mu_r \delta(B_r\{\tau_k\}) f^2(r) R(\tau_k) - \lambda_r f^3(r)\Big\}$$

$$=\sum_{r=1}^{3} E\Big\{3\lambda_r f(r) R^2(\tau_k) + 3\lambda_r f^2(r) R(\tau_k) - 3\mu_r \delta(B_r\{\tau_k\}) f(r) R^2(\tau_k)$$

$$+3\mu_r \delta(B_r\{\tau_k\}) f^2(r) R(\tau_k)\Big\}.$$

Now, expanding the above and removing a factor of three, we have, for $R(t)$ given in equation (4.11) that

$$
\begin{aligned}
0 \quad = \quad & E\Big\{\lambda_1 f(1)\left(f^2(1)N_1^2(\tau_k)+f^2(2)N_2^2(\tau_k)+f^2(3)N_3^2(\tau_k)\right.\\
& +2f(1)f(2)N_1(\tau_k)N_2(\tau_k)+2f(1)f(3)N_1(\tau_k)N_3(\tau_k)\\
& \left.+2f(2)f(3)N_2(\tau_k)N_3(\tau_k)\right)\\
& +\lambda_2 f(2)\left(f^2(1)N_1^2(\tau_k)+f^2(2)N_2^2(\tau_k)+f^2(3)N_3^2(\tau_k)\right.\\
& +2f(1)f(2)N_1(\tau_k)N_2(\tau_k)+2f(1)f(3)N_1(\tau_k)N_3(\tau_k)\\
& \left.+2f(2)f(3)N_2(\tau_k)N_3(\tau_k)\right)\\
& +\lambda_3 f(3)\left(f^2(1)N_1^2(\tau_k)+f^2(2)N_2^2(\tau_k)+f^2(3)N_3^2(\tau_k)\right.\\
& +2f(1)f(2)N_1(\tau_k)N_2(\tau_k)+2f(1)f(3)N_1(\tau_k)N_3(\tau_k)\\
& \left.+2f(2)f(3)N_2(\tau_k)N_3(\tau_k)\right)\\
& +\lambda_1 f^2(1)\left(f(1)N_1(\tau_k)+f(2)N_2(\tau_k)+f(3)N_3(\tau_k)\right)\\
& +\lambda_2 f^2(2)\left(f(1)N_1(\tau_k)+f(2)N_2(\tau_k)+f(3)N_3(\tau_k)\right)\\
& +\lambda_3 f^2(3)\left(f(1)N_1(\tau_k)+f(2)N_2(\tau_k)+f(3)N_3(\tau_k)\right)\\
& -\mu_1 f(1)\delta(B_1\{\tau_k\})\left(f^2(1)N_1^2(\tau_k)+f^2(2)N_2^2(\tau_k)+f^2(3)N_3^2(\tau_k)\right.\\
& +2f(1)f(2)N_1(\tau_k)N_2(\tau_k)+2f(1)f(3)N_1(\tau_k)N_3(\tau_k)\\
& \left.+2f(2)f(3)N_2(\tau_k)N_3(\tau_k)\right)\\
& -\mu_2 f(2)\delta(B_2\{\tau_k\})\left(f^2(1)N_1^2(\tau_k)+f^2(2)N_2^2(\tau_k)+f^2(3)N_3^2(\tau_k)\right.\\
& +2f(1)f(2)N_1(\tau_k)N_2(\tau_k)+2f(1)f(3)N_1(\tau_k)N_3(\tau_k)\\
& \left.+2f(2)f(3)N_2(\tau_k)N_3(\tau_k)\right)\\
& -\mu_3 f(3)\delta(B_3\{\tau_k\})\left(f^2(1)N_1^2(\tau_k)+f^2(2)N_2^2(\tau_k)+f^2(3)N_3^2(\tau_k)\right.\\
& +2f(1)f(2)N_1(\tau_k)N_2(\tau_k)+2f(1)f(3)N_1(\tau_k)N_3(\tau_k)\\
& \left.+2f(2)f(3)N_2(\tau_k)N_3(\tau_k)\right)\\
& +\mu_1 f^2(1)\delta(B_1\{\tau_k\})\left(f(1)N_1(\tau_k)+f(2)N_2(\tau_k)+f(3)N_3(\tau_k)\right)\\
& +\mu_2 f^2(2)\delta(B_2\{\tau_k\})\left(f(1)N_1(\tau_k)+f(2)N_2(\tau_k)+f(3)N_3(\tau_k)\right)\\
& +\mu_3 f^2(3)\delta(B_3\{\tau_k\})\left(f(1)N_1(\tau_k)+f(2)N_2(\tau_k)+f(3)N_3(\tau_k)\right)\Big\}.
\end{aligned}
$$

Now using the notation

$$n_r = E\left[N_r\left(\tau_k\right)\right] \tag{4.22}$$

$$M_{rs} = E\left[N_r\left(\tau_k\right)N_s\left(\tau_k\right)\right] \tag{4.23}$$

$$I_{rs} = E\left[\delta\left(B_r\left\{\tau_k\right\}\right)N_s\left(\tau_k\right)\right] \tag{4.24}$$

$$H_{rsw} = E\left[\delta\left(B_r\left\{\tau_k\right\}\right)N_s\left(\tau_k\right)N_w\left(\tau_k\right)\right] \text{ where } r,s,w=1,2,3 \tag{4.25}$$

and simplifying gives

$$\begin{aligned}
0 = \; & f^3(1)\left(\lambda_1 M_{11} + \mu_1 I_{11} + \lambda_1 n_1 - \mu_1 H_{111}\right) + f^3(2)\left(\lambda_2 M_{22} + \mu_2 I_{22} + \lambda_2 n_2 - \mu_2 H_{222}\right) \\
& + f^3(3)\left(\lambda_3 M_{33} + \mu_3 I_{33} + \lambda_3 n_3 - \mu_3 H_{333}\right) \\
& + f^2(1) f(2)\left(2\lambda_1 M_{12} - 2\mu_1 H_{112} - \mu_2 H_{211} + \mu_1 I_{12} + \lambda_1 n_2 + \lambda_2 M_{11}\right) \\
& + f^2(1) f(3)\left(2\lambda_1 M_{13} - 2\mu_1 H_{113} - \mu_3 H_{311} + \mu_1 I_{13} + \lambda_1 n_3 + \lambda_3 M_{11}\right) \\
& + f^2(2) f(1)\left(2\lambda_2 M_{21} - 2\mu_2 H_{221} - \mu_1 H_{122} + \mu_2 I_{21} + \lambda_2 n_1 + \lambda_1 M_{22}\right) \\
& + f^2(2) f(3)\left(2\lambda_2 M_{23} - 2\mu_2 H_{223} - \mu_3 H_{322} + \mu_2 I_{23} + \lambda_2 n_3 + \lambda_3 M_{22}\right) \\
& + f^2(3) f(1)\left(2\lambda_3 M_{31} - 2\mu_3 H_{331} - \mu_1 H_{133} + \mu_3 I_{31} + \lambda_3 n_1 + \lambda_1 M_{33}\right) \\
& + f^2(3) f(2)\left(2\lambda_3 M_{32} - 2\mu_3 H_{332} - \mu_2 H_{233} + \mu_3 I_{32} + \lambda_3 n_2 + \lambda_2 M_{33}\right) \\
& + f(1) f(2) f(3)\left(2\lambda_1 M_{23} + 2\lambda_2 M_{13} + 2\lambda_3 M_{12} - 2\mu_1 H_{123} - 2\mu_2 H_{213} - 2\mu_3 H_{312}\right).
\end{aligned}$$

Finally, equating the coefficients of $f(1)f(2)f(3)$, $f^2(3)f(2)$ etc., we develop the following set of identities

$$\lambda_1 M_{11} + \mu_1 I_{11} + \lambda_1 n_1 - \mu_1 H_{111} = 0 \tag{4.26}$$

$$\lambda_2 M_{22} + \mu_2 I_{22} + \lambda_2 n_2 - \mu_2 H_{222} = 0 \tag{4.27}$$

$$\lambda_3 M_{33} + \mu_3 I_{33} + \lambda_3 n_3 - \mu_3 H_{333} = 0 \tag{4.28}$$

$$2\lambda_1 M_{12} - 2\mu_1 H_{112} - \mu_2 H_{211} + \mu_1 I_{12} + \lambda_1 n_2 + \lambda_2 M_{11} = 0 \quad (4.29)$$

$$2\lambda_1 M_{13} - 2\mu_1 H_{113} - \mu_3 H_{311} + \mu_1 I_{13} + \lambda_1 n_3 + \lambda_3 M_{11} = 0 \quad (4.30)$$

$$2\lambda_2 M_{21} - 2\mu_2 H_{221} - \mu_1 H_{122} + \mu_2 I_{21} + \lambda_2 n_1 + \lambda_1 M_{22} = 0 \quad (4.31)$$

$$2\lambda_2 M_{23} - 2\mu_2 H_{223} - \mu_3 H_{322} + \mu_2 I_{23} + \lambda_2 n_3 + \lambda_3 M_{22} = 0 \quad (4.32)$$

$$2\lambda_3 M_{31} - 2\mu_3 H_{331} - \mu_1 H_{133} + \mu_3 I_{31} + \lambda_3 n_1 + \lambda_1 M_{33} = 0 \quad (4.33)$$

$$2\lambda_3 M_{32} - 2\mu_3 H_{332} - \mu_2 H_{233} + \mu_3 I_{32} + \lambda_3 n_2 + \lambda_2 M_{33} = 0 \quad (4.34)$$

$$2\lambda_1 M_{23} + 2\lambda_2 M_{13} + 2\lambda_3 M_{12} - 2\mu_1 H_{123} - 2\mu_2 H_{213} - 2\mu_3 H_{312} = 0 \quad (4.35)$$

All the required moments of N_1, N_2 and N_3 satisfy the sets of equations (4.16)-(4.21) and (4.26)-(4.35) just derived. Hence, the region defined by these equations relaxes the achievable performance region for the three customer system.

4.7 The semidefinite constraints on the three class problem.

We seek to solve the optimisation problem

$$min\,(c_1 n_1 + c_2 n_2 + c_3 n_3 + c_4 M_{11} + c_5 M_{22} + c_6 M_{33})$$

where the minimisation is over the set of achievable first and second moments. The sets derived using the potential function method relax the achievable performance region for the three customer system. Therefore we are now able to solve not the optimisation problem over the exact achievable space (which we do not have) but the related problem over a relaxation of the exact space, given by $\bigcap_{l=2}^{3} P_l$. The minimised cost over the latter will give a lower bound on the optimal cost for the problem. P_2 is defined by the set of equations (4.16) - (4.21) along with

$$\sum_{s=1}^{3} I_{1s} = n_1, \quad (4.36)$$

$$\sum_{r=1}^{3} I_{2r} = n_2, \tag{4.37}$$

$$\sum_{t=1}^{3} I_{3t} = n_3, \tag{4.38}$$

$$I_{rs} \geq 0, \;\; r = 1,2,3 \;\; s = 1,2,3 \;, \tag{4.39}$$

$$n_r \geq 0, \;\; r = 1,2,3. \tag{4.40}$$

P_3 is given by (4.26)-(4.35) along with

$$\sum_{r=1}^{3} H_{r11} - M_{11} = 0, \tag{4.41}$$

$$\sum_{r=1}^{3} H_{r22} - M_{22} = 0, \tag{4.42}$$

$$\sum_{r=1}^{3} H_{r33} - M_{33} = 0, \tag{4.43}$$

$$\sum_{r=1}^{3} H_{r12} - M_{12} = 0, \tag{4.44}$$

$$\sum_{r=1}^{3} H_{r23} - M_{23} = 0, \tag{4.45}$$

$$\sum_{r=1}^{3} H_{r13} - M_{13} = 0, \tag{4.46}$$

$$n_r, M_{rs}, I_{rs}, H_{rst} \geq 0. \tag{4.47}$$

As in Chapter 2, we use the semidefinite programming methods of Vandenberghe & Boyd (1996). To obtain a lower bound on the cost achievable under any policy for our three class system, we wish to solve the following linear programming problem

$$\text{minimise } \mathbf{c}^T\mathbf{x},$$

$$\text{subject to } A\mathbf{x} + \mathbf{b} \geq \mathbf{0}$$

where $A = [a_1, a_2, \ldots, a_m] \in \mathbb{R}^{n \times m}$, $\mathbf{c} \in \mathbb{R}^m$ is a vector and $\mathbf{b} \in \mathbb{R}^n$ also a vector.

We denote $\mathbf{diag}\,(\mathbf{v})$ as the diagonal matrix having the components of $\mathbf{v}$ on the diagonal. We can state that a vector $\mathbf{v} \geq \mathbf{0}$ if and only if the matrix $\mathbf{diag}\,(\mathbf{v})$ is positive semidefinite.

We now rewrite the linear program above as the following semidefinite programming problem:

$$\text{minimise } \mathbf{c}^T\mathbf{x},$$

$$\text{subject to } F(x) \succeq \mathbf{0}$$

where $F(x) \equiv F_0 + \sum_{i=1}^{36} x_i F_i$, and $F_0 = \mathbf{diag}(\mathrm{b})$ and $F_i = \mathbf{diag}(\mathrm{a}_i), i = 1, 2, \ldots, 36$.

We note that $\mathbf{diag}(\mathrm{a})$ and $\mathbf{diag}(\mathrm{b})$ are of the form $\mathbf{diag}(\mathrm{v})$ above.

The additional semidefinite constraints, suggested by Bertsimas & Niño-Mora (1996), D_1, D_2, D_3 and D_4, are as follows

$$D_1 = \left\{ \begin{array}{cccc} 1 & n_1 & n_2 & n_3 \\ n_1 & M_{11} & M_{12} & M_{13} \\ n_2 & M_{12} & M_{22} & M_{23} \\ n_3 & M_{13} & M_{23} & M_{33} \end{array} \right\} \succeq 0 \quad (4.48)$$

$$D_2 = \left\{ \begin{array}{cccc} 1 & I_{11} & I_{12} & I_{13} \\ I_{11} & H_{111} & H_{112} & H_{113} \\ I_{12} & H_{112} & H_{122} & H_{123} \\ I_{13} & H_{113} & H_{123} & H_{133} \end{array} \right\} \succeq 0 \quad (4.49)$$

$$D_3 = \left\{ \begin{array}{cccc} 1 & I_{21} & I_{22} & I_{23} \\ I_{21} & H_{211} & H_{212} & H_{213} \\ I_{22} & H_{212} & H_{222} & H_{223} \\ I_{23} & H_{213} & H_{223} & H_{233} \end{array} \right\} \succeq 0 \quad (4.50)$$

$$D_4 = \left\{ \begin{array}{cccc} 1 & I_{31} & I_{32} & I_{33} \\ I_{31} & H_{311} & H_{312} & H_{313} \\ I_{32} & H_{312} & H_{322} & H_{323} \\ I_{33} & H_{313} & H_{323} & H_{333} \end{array} \right\} \succeq 0 \quad (4.51)$$

It only remains for us to reformulate the set $P_2 \bigcap P_3$ along with the additional semidefinite constraints of (4.48)-(4.51) into the semidefinite program. The non-parametric bounding method produced constraints of the form $A_1\mathbf{x} - \mathbf{b}_1 = \mathbf{0}$ while the standard form requires them to be in the form $A_1\mathbf{x} - \mathbf{b}_1 \geq \mathbf{0}$. Thus, in order to achieve this required form, we write the constraints as follows:

$$A_1\mathbf{x} \geq \mathbf{b}_1$$

$$-A_1\mathbf{x} \geq -\mathbf{b}_1$$

where $A_1 = (A_i, A_{ii}, A_{iii}, A_{iv})$ is a 25 by 36 matrix made up as indicated by the following:

$$A_i = \begin{pmatrix}
-\lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\lambda_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -\lambda_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\lambda_2 & -\lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\lambda_3 & -\lambda_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\lambda_3 & 0 & -\lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\lambda_1 & 0 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & 0 \\
0 & \lambda_2 & 0 & 0 & 0 & 0 & \lambda_2 & 0 & 0 \\
0 & 0 & \lambda_3 & 0 & 0 & 0 & 0 & 0 & \lambda_3 \\
0 & \lambda_1 & 0 & \lambda_2 & 2\lambda_1 & 0 & 0 & 0 & 0 \\
0 & 0 & \lambda_1 & \lambda_3 & 0 & 2\lambda_1 & 0 & 0 & 0 \\
\lambda_2 & 0 & 0 & 0 & 2\lambda_2 & 0 & \lambda_1 & 0 & 0 \\
\lambda_3 & 0 & 0 & 0 & 0 & 2\lambda_3 & 0 & 0 & \lambda_1 \\
0 & 0 & \lambda_2 & 0 & 0 & 0 & \lambda_3 & 2\lambda_2 & 0 \\
0 & \lambda_3 & 0 & 0 & 0 & 0 & 0 & 2\lambda_3 & \lambda_2 \\
0 & 0 & 0 & 0 & 2\lambda_3 & 2\lambda_2 & 0 & 2\lambda_1 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1
\end{pmatrix}
\quad
A_{iii} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\mu_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -2\mu_1 & 0 & 0 & 0 & 0 & -\mu_2 & 0 & 0 \\
0 & 0 & -2\mu_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\mu_1 & 0 & 0 & 0 & -2\mu_2 & 0 \\
0 & 0 & 0 & 0 & 0 & -\mu_1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -2\mu_1 & 0 & 0 & 0 & -2\mu_2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{pmatrix}$$

$$
A_{ii} = \left\{\begin{array}{ccccccccc}
\mu_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mu_2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_3 \\
0 & \mu_1 & 0 & \mu_2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \mu_2 & 0 & \mu_3 & 0 \\
0 & 0 & \mu_1 & 0 & 0 & 0 & \mu_3 & 0 & 0 \\
\mu_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \mu_2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_3 \\
0 & \mu_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \mu_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mu_2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mu_3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \mu_2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_3 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right\}
\qquad
A_{iv} = \left\{\begin{array}{ccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\mu_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\mu_3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\mu_3 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -2\mu_3 & 0 & 0 & 0 \\
0 & -2\mu_2 & 0 & 0 & 0 & 0 & -\mu_3 & 0 & 0 \\
0 & 0 & -\mu_2 & 0 & 0 & 0 & 0 & -2\mu_3 & 0 \\
0 & 0 & 0 & 0 & -2\mu_3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right\}.
$$

In addition

$$
\begin{aligned}
\mathbf{x}^T = \big(& n_1, n_2, n_3, M_{11}, M_{12}, M_{13}, M_{22}, M_{23}, M_{33}, I_{11}, I_{12}, I_{13}, I_{21}, \\
& I_{22}, I_{23}, I_{31}, I_{32}, I_{33}, H_{111}, H_{112}, H_{113}, H_{122}, H_{123}, H_{133}, H_{211}, \\
& H_{212}, H_{213}, H_{222}, H_{223}, H_{233}, H_{311}, H_{312}, H_{313}, H_{322}, H_{323}, H_{333}\big).
\end{aligned}
$$

E again takes the form $E \geq \mathbf{0}$, where E now represents a 36 by 36 identity matrix and

$$
\mathbf{b}^T = \big(-\mathbf{b}_1^T, \mathbf{b}_1^T, 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\big) \tag{4.52}
$$

where

$$
\mathbf{b}_1^T = (\lambda_1, \lambda_2, \lambda_3, 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
$$

We can now write all of the linear constraints in the required form: $A\mathbf{x} - \mathbf{b} \geq \mathbf{0}$ where

$$
A = \left\{\begin{array}{c} -A_1 \\ A_1 \\ E \end{array}\right\}
$$

and b is the transpose of (4.52). We are now able to formulate the problem as a semidefinite program

$$\text{minimise} \quad c_1 n_1 + c_2 n_2 + c_3 n_3 + c_4 M_{11} + c_5 M_{22} + c_6 M_{33}$$

$$\text{subject to} \quad \left\{ \begin{array}{ccccc} \operatorname{diag}(A\mathrm{x} - \mathrm{b}) & 0 & 0 & 0 & 0 \\ 0 & D_1 & 0 & 0 & 0 \\ 0 & 0 & D_2 & 0 & 0 \\ 0 & 0 & 0 & D_3 & 0 \\ 0 & 0 & 0 & 0 & D_4 \end{array} \right\} \succeq 0$$

The semidefinite program constraints for the above problem can be written in the form

$$\sum_{i=1}^{36} x_i F_i - F_0 \succeq 0$$

and then be solved. Again, we used the software developed by Kojima (1994)

4.8 Calculation of C^{OPT} for the three class problem

As in the two customer case, we calculate C^{OPT} using the method of dynamic programming as described in Tijms (1994). The recursion used in the three class problem is given below.

$$\begin{aligned} C_{t+1}(n_1, n_2, n_3) \;=\; \min \Bigg\{ & \frac{b_1 n_1 + b_2 n_2 + b_3 n_3 + c_1 n_1^2 + c_2 n_2^2 + c_3 n_3^2}{\lambda_1 + \lambda_2 + \lambda_3 + \mu_1} \\ & + \frac{\lambda_1 C_t(n_1 + 1, n_2, n_3) + \lambda_2 C_t(n_1, n_2 + 1, n_3) + \lambda_3 C_t(n_1, n_2, n_3 + 1) + \mu_1 C_t(n_1 - 1, n_2, n_3)}{\lambda_1 + \lambda_2 + \lambda_3 + \mu_1}; \\ & \frac{b_1 n_1 + b_2 n_2 + b_3 n_3 + c_1 n_1^2 + c_2 n_2^2 + c_3 n_3^2}{\lambda_1 + \lambda_2 + \lambda_3 + \mu_2} \\ & + \frac{\lambda_1 C_t(n_1 + 1, n_2, n_3) + \lambda_2 C_t(n_1, n_2 + 1, n_3) + \lambda_3 C_t(n_1, n_2, n_3 + 1) + \mu_2 C_t(n_1, n_2 - 1, n_3)}{\lambda_1 + \lambda_2 + \lambda_3 + \mu_2}; \\ & \frac{b_1 n_1 + b_2 n_2 + b_3 n_3 + c_1 n_1^2 + c_2 n_2^2 + c_3 n_3^2}{\lambda_1 + \lambda_2 + \lambda_3 + \mu_3} \\ & + \frac{\lambda_1 C_t(n_1 + 1, n_2, n_3) + \lambda_2 C_t(n_1, n_2 + 1, n_3) + \lambda_3 C_t(n_1, n_2, n_3 + 1) + \mu_3 C_t(n_1, n_2, n_3 - 1)}{\lambda_1 + \lambda_2 + \lambda_3 + \mu_3} \Bigg\} \\ & (n_1, n_2, n_3) \in \left(\mathbb{Z}^+\right)^3 \end{aligned}$$

$$
\begin{aligned}
C_{t+1}(n_1,n_2,0) &= \min\left\{\frac{b_1n_1+b_2n_2+c_1n_1^2+c_2n_2^2}{\lambda_1+\lambda_2+\lambda_3+\mu_1}\right.\\
&\quad+\frac{\lambda_1C_t(n_1+1,n_2,0)+\lambda_2C_t(n_1,n_2+1,0)+\lambda_3C_t(n_1,n_2,1)+\mu_1C_t(n_1-1,n_2,0)}{\lambda_1+\lambda_2+\lambda_3+\mu_1};\\
&\quad\frac{b_1n_1+b_2n_2+c_1n_1^2+c_2n_2^2}{\lambda_1+\lambda_2+\lambda_3+\mu_2}\\
&\quad\left.+\frac{\lambda_1C_t(n_1+1,n_2,0)+\lambda_2C_t(n_1,n_2+1,0)+\lambda_3C_t(n_1,n_2,1)+\mu_2C_t(n_1,n_2-1,0)}{\lambda_1+\lambda_2+\lambda_3+\mu_2}\right\}\\
&\qquad (n_1,n_2)\in\left(\mathbb{Z}^+\right)^2
\end{aligned}
$$

$$
\begin{aligned}
C_{t+1}(n_1,0,n_3) &= \min\left\{\frac{b_1n_1+b_3n_3+c_1n_1^2+c_3n_3^2}{\lambda_1+\lambda_2+\lambda_3+\mu_1}\right.\\
&\quad+\frac{\lambda_1C_t(n_1+1,0,n_3)+\lambda_2C_t(n_1,1,n_3)+\lambda_3C_t(n_1,0,n_3+1)+\mu_1C_t(n_1-1,0,n_3)}{\lambda_1+\lambda_2+\lambda_3+\mu_1};\\
&\quad\frac{b_1n_1+b_3n_3+c_1n_1^2+c_3n_3^2}{\lambda_1+\lambda_2+\lambda_3+\mu_1}\\
&\quad\left.+\frac{\lambda_1C_t(n_1+1,0,n_3)+\lambda_2C_t(n_1,1,n_3)+\lambda_3C_t(n_1,0,n_3+1)+\mu_3C_t(n_1,0,n_3-1)}{\lambda_1+\lambda_2+\lambda_3+\mu_1}\right\}\\
&\qquad (n_1,n_3)\in\left(\mathbb{Z}^+\right)^2
\end{aligned}
$$

$$
\begin{aligned}
C_{t+1}(0,n_2,n_3) &= \min\left\{\frac{b_2n_2+b_3n_3+c_2n_2^2+c_3n_3^2}{\lambda_1+\lambda_2+\lambda_3+\mu_2}\right.\\
&\quad+\frac{\lambda_1C_t(1,n_2,n_3)+\lambda_2C_t(0,n_2+1,n_3)+\lambda_3C_t(0,n_2,n_3+1)+\mu_2C_t(0,n_2-1,n_3)}{\lambda_1+\lambda_2+\lambda_3+\mu_2};\\
&\quad\frac{b_2n_2+b_3n_3+c_2n_2^2+c_3n_3^2}{\lambda_1+\lambda_2+\lambda_3+\mu_3}\\
&\quad\left.+\frac{\lambda_1C_t(1,n_2,n_3)+\lambda_2C_t(0,n_2+1,n_3)+\lambda_3C_t(0,n_2,n_3+1)+\mu_3C_t(0,n_2,n_3-1)}{\lambda_1+\lambda_2+\lambda_3+\mu_3}\right\}\\
&\qquad (n_2,n_3)\in\left(\mathbb{Z}^+\right)^2
\end{aligned}
$$

$$
\begin{aligned}
(\lambda_1+\lambda_2+\lambda_3+\mu_1)\,C_{t+1}(n_1,0,0) &= b_1n_1+c_1n_1^2+\lambda_1C_t(n_1+1,0,0)+\lambda_2C_t(n_1,1,0)\\
&\quad+\lambda_3C_t(n_1,0,1)+\mu_1C_t(n_1-1,0,0),\ n_1\in\mathbb{Z}^+
\end{aligned}
$$

$$
\begin{aligned}
(\lambda_1+\lambda_2+\lambda_3+\mu_2)\,C_{t+1}(0,n_2,0) &= b_2n_2+c_2n_2^2+\lambda_1C_t(1,n_2,0)+\lambda_2C_t(0,n_2+1,0)\\
&\quad+\lambda_3C_t(0,n_2,1)+\mu_2C_t(0,n_2-1,0),\ n_2\in\mathbb{Z}^+
\end{aligned}
$$

$$
\begin{aligned}
(\lambda_1+\lambda_2+\lambda_3+\mu_3)\,C_{t+1}(0,0,n_3) &= b_2n_3+c_3n_3^2+\lambda_1C_t(1,0,n_3)+\lambda_2C_t(0,1,n_3)\\
&\quad+\lambda_3C_t(0,0,n_3+1)+\mu_3C_t(0,0,n_3-1),\ n_3\in\mathbb{Z}^+
\end{aligned}
$$

$$(\lambda_1 + \lambda_2 + \lambda_3) C_{t+1}(0,0,0) = \lambda_1 C_t(1,0,0) + \lambda_2 C_t(0,1,0) + \lambda_3 C_t(0,0,1)$$

$$t \in \mathbb{N}$$

Our calculations were computed over a state space large enough to give a result with $\epsilon = 0.000000001$, the required degree of accuracy as given in the value-iteration algorithm in Table 4.1. It is likely that this method would not be a realistic possibility for larger problems and indeed the calculations of results for the three class problems did take considerably longer to compute than those for two. Therefore, in our further investigations of section 4.9.1, we decided to measure index policy performance against the lower bound cost only. This was because it was felt that this will have to be the method in future work dealing with larger problem instances and that, as will be seen, in all of the problems where C^{OPT} was calculated, the index policy cost was close to it and usually within 10% of the lower bound. In the results of section 4.9.1, the index policy costs were also usually within 10% of the lower bound and this was interpreted as a good performance.

4.9 Numerical examples of three class type problems

Tables (4.4)-(4.7) show the costs achieved by the Whittle index policy on three class M/M/1 systems with arrival rates and service rates as indicated. The service rates were randomly generated on the interval (0.1, 20.0) and the arrival rates were calculated by scaling three further numbers randomly generated on the same interval so that ρ was fixed at 0.75. In Tables 4.4-4.7, the cost coefficients are taken to be $b_1 = 5$, $b_2 = 1$, $b_3 = 1$, $c_1 = 1$, $c_2 = 2$ and $c_3 = 0.2$. We define C^{IND} as the cost incurred following the Whittle index policy; C^{OPT} as the minimum achievable cost, calculated by means of dynamic programming and C^{SD} as the semidefinite lower bound on the minimum achievable cost. Further, we define $\%^{OPT}$ as the percentage increase in C^{IND} over C^{OPT}; and $\%^{SD}$ as the percentage increase in C^{IND} over C^{SD}.

Results show that the Whittle index policies perform close to optimally in all cases. In Tables 4.4 - 4.7 the values of $\%^{OPT}$ range from 0.000 to 0.712. The great majority of values are close to zero and this is indicated from the median, 0.000, and the lower and upper quartiles, 0.000 and 0.019 respectively. The values of $\%^{SD}$ range from 0.060 to 11.541. Again, the values are skewed and this is indicated from the median, 2.789, and the lower and upper quartiles, 1.124 and 5.935.

Table 4.4: Results for Whittle index policies in three customer type problems.

λ_1	λ_2	λ_3	μ_1	μ_2	μ_3	C^{IND}	C^{OPT}	C^{SD}	$\%^{OPT}$	$\%^{SD}$
2.384	4.166	0.726	13.535	12.761	2.934	7.827	7.824	7.481	0.038	4.625
6.592	3.657	1.485	14.538	17.290	17.459	16.531	16.511	15.614	0.121	5.873
3.338	0.251	2.203	12.896	0.953	9.681	11.713	11.708	11.198	0.043	4.599
2.878	0.998	1.107	12.433	5.410	3.313	6.950	6.950	6.647	0.000	4.558
1.825	1.113	3.194	17.273	13.964	5.657	6.451	6.450	6.337	0.016	1.799
2.492	3.473	2.828	17.318	18.388	6.777	6.464	6.463	6.248	0.015	3.457
0.106	0.065	0.091	6.981	16.455	0.124	7.035	7.035	7.034	0.000	0.014
0.663	1.186	4.041	17.301	15.802	6.347	6.619	6.619	6.517	0.000	1.565
1.066	1.405	0.250	1.848	13.922	3.468	20.542	20.542	20.235	0.000	1.517
0.290	0.086	0.304	0.427	3.000	7.080	31.430	31.412	31.208	0.057	0.711
0.999	1.526	3.331	16.555	3.566	12.731	16.816	16.773	16.256	0.256	3.445
0.618	0.987	0.880	10.804	2.798	2.586	8.335	8.335	8.049	0.000	3.553
0.871	0.877	0.880	2.253	19.911	2.757	10.677	10.677	10.179	0.000	4.892
2.774	2.685	1.673	13.033	16.360	4.484	6.653	6.653	6.411	0.000	3.775
0.434	1.983	0.840	10.756	6.537	2.067	6.271	6.271	6.165	0.000	1.719
2.070	0.597	1.999	4.090	4.456	18.181	20.848	20.791	19.138	0.274	8.935
0.316	0.538	0.348	15.298	13.816	0.504	6.647	6.647	6.643	0.000	0.060
0.708	0.236	0.495	1.317	1.260	19.622	24.240	24.225	22.338	0.062	8.515
0.759	1.024	1.076	17.347	2.024	5.376	17.756	17.739	17.143	0.096	3.576
4.612	2.529	5.926	15.325	16.762	19.878	10.478	10.478	9.503	0.000	10.260
1.104	0.506	2.333	12.723	12.245	3.751	6.578	6.578	6.507	0.000	1.091
1.738	1.976	1.026	14.813	11.160	2.251	5.868	5.868	5.729	0.000	2.426
1.318	0.314	0.410	18.652	0.478	17.565	36.076	36.072	35.913	0.011	0.454
1.680	1.964	0.136	2.654	18.027	16.861	28.730	28.728	28.549	0.007	0.634

Table 4.5: Results for Whittle index policies in three customer type problems contd.

λ_1	λ_2	λ_3	μ_1	μ_2	μ_3	C^{IND}	C^{OPT}	C^{SD}	$\%^{OPT}$	$\%^{SD}$
0.705	1.013	1.701	6.475	4.984	3.886	7.228	7.228	6.871	0.000	5.196
0.522	1.398	0.744	1.874	3.772	7.375	16.827	16.827	16.392	0.000	2.654
0.264	0.095	0.133	0.526	5.985	0.574	12.895	12.895	12.581	0.000	2.496
2.822	0.646	2.732	15.550	19.838	5.097	6.584	6.584	6.497	0.000	1.339
0.262	0.180	0.339	4.932	16.794	0.494	6.745	6.745	6.738	0.000	0.104
3.455	6.236	0.931	14.757	14.740	10.023	14.907	14.907	14.618	0.000	1.977
1.419	5.038	2.908	4.976	15.925	19.591	14.570	14.565	14.291	0.034	1.952
2.956	0.760	3.391	8.672	16.209	9.362	9.636	9.636	9.035	0.000	6.652
0.053	0.193	0.100	10.350	0.264	7.135	43.010	42.987	42.773	0.054	0.554
1.240	5.292	4.835	15.583	15.504	14.689	8.831	8.831	8.371	0.000	5.495
0.543	7.706	3.179	10.765	15.827	14.951	11.479	11.479	11.144	0.000	3.006
1.303	5.456	0.263	10.174	9.046	13.979	25.754	25.572	25.374	0.712	1.498
1.754	0.098	1.770	4.132	1.919	6.452	11.764	11.762	10.653	0.017	10.429
1.101	0.890	0.719	14.128	5.803	1.386	5.881	5.881	5.815	0.000	1.135
0.373	4.160	2.657	12.324	12.322	6.953	7.180	7.180	7.016	0.000	2.338
0.287	0.231	0.570	0.443	5.686	9.204	29.468	29.464	29.326	0.014	0.484
0.367	0.431	0.551	0.579	19.190	5.941	28.592	28.555	28.301	0.130	1.028
0.635	0.872	0.043	1.115	5.013	6.313	26.030	26.023	25.840	0.027	0.735
0.650	0.868	0.589	17.084	8.731	0.962	6.110	6.110	6.091	0.000	0.312
1.334	5.483	3.875	9.907	13.983	17.361	11.700	11.700	11.021	0.000	6.161
0.202	1.078	0.853	3.113	14.410	1.397	6.397	6.396	6.315	0.016	1.298
0.409	2.873	1.100	2.594	5.604	13.833	17.110	17.109	16.720	0.006	2.333
0.528	0.129	0.733	3.425	15.403	1.247	6.749	6.749	6.694	0.000	0.822

Table 4.6: Results for Whittle index policies in three customer type problems contd.

λ_1	λ_2	λ_3	μ_1	μ_2	μ_3	C^{IND}	C^{OPT}	C^{SD}	$\%^{OPT}$	$\%^{SD}$
1.179	0.510	0.961	11.484	19.520	1.547	6.423	6.423	6.398	0.000	0.391
1.919	2.074	0.858	15.488	4.771	4.486	11.233	11.232	10.632	0.009	5.653
0.698	0.321	0.791	15.016	0.518	9.323	32.618	32.584	32.198	0.104	1.304
0.557	0.308	0.049	0.770	13.037	13.893	34.211	34.206	34.128	0.015	0.243
7.235	1.628	0.264	14.401	6.954	19.689	19.899	19.874	17.840	0.126	11.541
0.179	0.340	0.122	6.364	13.312	0.175	6.721	6.721	6.716	0.000	0.074
1.350	2.131	4.567	12.891	14.646	9.137	6.968	6.968	6.642	0.000	4.908
2.293	1.621	7.215	12.270	11.718	16.987	9.222	9.221	8.476	0.011	8.801
7.359	0.130	1.434	11.831	13.319	12.138	18.090	18.090	17.837	0.000	1.418
3.193	2.685	3.326	17.326	10.497	10.733	8.558	8.557	7.958	0.012	7.540
0.172	0.222	0.018	1.988	0.337	2.973	36.486	36.471	35.547	0.041	2.642
9.041	2.459	0.052	19.880	8.412	17.474	21.090	21.090	18.992	0.000	11.047
2.074	2.200	0.836	11.635	18.351	1.849	5.997	5.997	5.873	0.000	2.111
9.692	0.584	0.838	18.501	3.217	18.820	15.054	15.054	14.667	0.000	2.639
2.038	1.652	1.212	10.866	18.290	2.567	6.216	6.216	6.094	0.000	2.002
1.356	3.650	5.530	19.815	12.224	14.440	8.723	8.723	8.220	0.000	6.119
5.483	3.353	1.529	15.982	14.244	8.916	10.730	10.730	9.880	0.000	8.603
3.323	1.158	3.515	15.484	18.183	7.451	6.900	6.900	6.704	0.000	2.924
5.036	1.651	4.910	14.142	15.107	17.252	10.725	10.725	9.819	0.000	9.227
2.991	2.227	0.186	9.013	5.551	10.969	23.072	23.069	21.289	0.013	8.375
8.460	0.134	4.517	19.117	2.611	17.623	9.980	9.979	9.690	0.010	2.993
0.510	0.165	0.381	1.182	2.770	1.466	11.889	11.888	11.195	0.008	6.199
0.363	0.503	0.620	6.044	7.422	0.997	6.311	6.311	6.275	0.000	0.574

Table 4.7: Results for Whittle index policies in three customer type problems contd.

λ_1	λ_2	λ_3	μ_1	μ_2	μ_3	C^{IND}	C^{OPT}	C^{SD}	$\%^{OPT}$	$\%^{SD}$
4.911	4.057	2.366	17.953	14.721	11.775	10.172	10.171	9.388	0.010	8.351
0.267	2.670	3.304	11.783	10.167	7.109	7.035	7.035	6.831	0.000	2.986
1.735	6.786	0.217	17.754	10.883	7.565	26.906	26.855	26.620	0.190	1.074
0.295	0.512	0.625	5.712	0.849	6.578	30.514	30.410	29.430	0.342	3.683
0.143	3.062	2.263	14.419	13.252	4.445	6.113	6.113	6.050	0.000	1.041
3.186	3.190	3.992	10.130	16.228	16.709	11.975	11.975	11.030	0.000	8.568
3.012	0.042	3.021	16.024	14.231	5.403	6.759	6.759	6.699	0.000	0.896
3.314	3.233	1.273	12.038	14.858	4.950	8.132	8.132	7.746	0.000	4.983
0.211	0.496	0.541	0.450	9.915	2.346	19.601	19.555	19.454	0.235	0.756
2.142	1.044	2.856	10.511	2.963	14.734	16.212	16.197	14.963	0.093	8.347
0.487	0.793	0.286	1.031	3.044	16.743	23.240	23.234	22.969	0.026	1.180
5.496	2.124	2.636	13.988	15.184	12.140	10.732	10.731	9.928	0.009	8.098
1.602	3.721	4.742	18.832	14.738	11.497	7.396	7.396	7.044	0.000	4.997
4.458	3.608	4.204	19.096	17.165	13.724	8.438	8.437	7.852	0.012	7.463
4.200	0.730	4.044	13.327	6.313	12.670	9.459	9.459	8.814	0.000	7.318
0.561	0.383	1.557	3.794	14.285	2.707	7.159	7.159	6.943	0.000	3.111
1.829	1.786	1.139	3.417	19.168	9.369	19.783	19.780	18.794	0.015	5.262
1.772	2.005	2.721	10.362	6.813	9.557	10.204	10.197	9.338	0.069	9.274
6.291	0.290	1.569	12.548	10.422	7.107	11.157	11.157	10.932	0.000	2.058
0.287	1.155	0.946	0.558	11.122	7.173	22.006	22.004	21.818	0.009	0.862
2.438	0.044	0.859	7.749	6.893	2.001	7.031	7.031	6.990	0.000	0.587

4.9.1 Further investigations

Seeking to investigate further the performance of the Whittle index policies, with a number of different values of cost coefficients, we decided to select a small number of the above systems and to allow the cost coefficients to vary. Initially, we selected the 4 systems shown below as they appear in Tables 4.4 - 4.7 and as they seemed to us to be fairly representative of the range of index policy performances(from those calculated) i.e. the index costs in the four selected systems range from 0.014% to 11.047% above the relevant SDLB costs. The results are shown in Tables (4.9) - (4.12).

Table 4.8: Systems selected for further investigation.

λ_1	λ_2	λ_3	μ_1	μ_2	μ_3	C^{IND}	C^{OPT}	C^{SD}	$\%^{OPT}$	$\%^{SD}$
1.240	5.292	4.835	15.583	15.504	14.689	8.831	8.831	8.371	0.000	5.495
0.106	0.065	0.091	6.981	16.455	0.124	7.035	7.035	7.034	0.000	0.014
9.041	2.459	0.052	19.880	8.412	17.474	21.090	21.090	18.992	0.000	11.047
1.101	0.890	0.719	14.128	5.803	1.386	5.881	5.881	5.815	0.000	1.135

In these further investigations, the values of the b_i, $i = 1, 2, 3$ are kept constant while we allow the second moment coefficients to vary.

Table 4.9: Results for system with $\lambda_1 = 1.24$, $\lambda_2 = 5.292$, $\lambda_3 = 4.835$, $\mu_1 = 15.583$, $\mu_2 = 15.504$ and $\mu_3 = 14.689$

b_1	b_2	b_3	c_1	c_2	c_3	C^{IND}	C^{SD}	$\%^{SD}$
5	1	1	0.00	0.00	0.00	3.294	3.294	0.000
5	1	1	0.01	0.02	0.002	3.353	3.345	0.239
5	1	1	0.10	0.20	0.02	3.879	3.802	2.025
5	1	1	0.20	0.40	0.04	4.462	4.310	3.527
5	1	1	0.30	0.60	0.06	5.035	4.817	4.526
5	1	1	0.40	0.80	0.08	5.598	5.325	5.127
5	1	1	1.00	2.00	0.20	8.831	8.371	5.495
5	1	1	5.00	10.00	1.00	29.119	27.317	6.597
5	1	1	10.00	20.00	2.00	54.079	50.850	6.350

In Table 4.9 the results obtained were comparable to those of Tables (4.4) - (4.7) and we felt that the semidefinite lower bound cost was sufficiently tight to assess performance. The $\%^{SD}$ tended to increase as the values of the c_i, $i = 1, 2, 3$ increased. The largest value of $\%^{SD}$ occurred when $c_1 = 10.00$, $c_2 = 20.00$ and $c_1 = 2.00$ but this only represented an increase of 6.35 %. Thus the bounds appear to be tight and the index policies perform well.

Table 4.10: Results for system with $\lambda_1 = 0.106,\ \lambda_2 = 0.065,\ \lambda_3 = 0.091,\ \mu_1 = 6.981,\ \mu_2 = 16.455$ and $\mu_3 = 0.124$

b_1	b_2	b_3	c_1	c_2	c_3	C^{IND}	C^{SD}	$\%^{SD}$
5	1	1	0.00	0.00	0.000	3.006	3.006	0.000
5	1	1	0.01	0.02	0.002	3.046	3.046	0.000
5	1	1	0.10	0.20	0.020	3.409	3.409	0.000
5	1	1	0.20	0.40	0.040	3.812	3.811	0.026
5	1	1	0.30	0.60	0.060	4.215	4.214	0.024
5	1	1	0.40	0.80	0.080	4.617	4.617	0.000
5	1	1	1.00	2.00	0.200	7.035	7.034	0.014
5	1	1	5.00	10.00	1.000	23.152	23.150	0.009
5	1	1	10.00	20.00	2.000	43.298	43.295	0.007

Table (4.10), shows the results for a set of problems where arrival and service rates are as indicated but the cost coefficients are the same as those of the results in Table (4.9). Here the values $\%^{SD}$ range from 0 to 0.026 and it is clear that the index policies are performing at close to optimal for all values of c_1, c_2 and c_3 investigated.

Table 4.11: Results for system with $\lambda_1 = 9.041$, $\lambda_2 = 2.459$, $\lambda_3 = 0.052$, $\mu_1 = 19.880$, $\mu_2 = 8.412$ and $\mu_3 = 17.474$

b_1	b_2	b_3	c_1	c_2	c_3	C^{IND}	C^{SD}	$\%^{SD}$
5	1	1	0.00	0.00	0.000	5.767	5.767	0.000
5	1	1	0.01	0.02	0.002	5.929	5.900	0.491
5	1	1	0.10	0.20	0.020	7.373	7.102	0.382
5	1	1	0.20	0.40	0.040	8.960	8.432	6.262
5	1	1	0.30	0.60	0.060	10.540	9.752	8.080
5	1	1	0.40	0.80	0.080	12.089	11.072	9.185
5	1	1	1.00	2.00	0.200	21.090	18.992	11.047
5	1	1	5.00	10.00	1.000	77.802	71.790	8.374
5	1	1	10.00	20.00	2.000	148.139	137.790	7.511

Table (4.11), shows the results for a set of problems where arrival and service rates are as indicated but again the cost coefficients are the same as those of the results in Table (4.9). Here the values $\%^{SD}$ range from 0 to 11.047. Once again, these results are comparable to those of Tables (4.4) - (4.7) and we infer from this that the index policy is performing well. We continued our investigations by further analysis on the system with $\lambda_1 = 0.106$, $\lambda_2 = 0.065$, $\lambda_3 = 0.091$, $\mu_1 = 6.981$, $\mu_2 = 16.455$ and $\mu_3 = 0.124$ as the index based cost was extremely close to the SDLB cost for all coefficient values. We now used further variations on the cost coefficients. Results are also given for the system where $\lambda_1 = 1.101$, $\lambda_2 = 0.890$, $\lambda_3 = 0.719$, $\mu_1 = 14.128$, $\mu_2 = 5.803$ and $\mu_3 = 1.386$ using these new coefficients as this was the next best performing system of those in Table 4.8, i.e. $\%^{SD} = 1.135$.

Table 4.12: Results for system with $\lambda_1 = 0.106$, $\lambda_2 = 0.065$, $\lambda_3 = 0.091$, $\mu_1 = 6.981$, $\mu_2 = 16.455$ and $\mu_3 = 0.124$

b_1	b_2	b_3	c_1	c_2	c_3	C^{IND}	C^{SD}	$\%^{SD}$
5	2	1	0.00	0.00	0.000	3.010	3.010	0.000
5	2	1	0.01	0.02	0.050	4.011	4.010	0.025
5	2	1	0.02	0.04	0.100	5.013	5.011	0.040
5	2	1	0.05	0.02	0.010	3.211	3.211	0.000
5	2	1	0.05	0.10	0.250	8.017	8.012	0.062
5	2	1	0.10	0.10	0.100	5.014	5.014	0.000
5	2	1	0.10	0.20	0.500	13.025	13.015	0.077
5	2	1	0.20	0.20	0.200	7.019	7.018	0.014
5	2	1	0.20	0.40	1.000	23.040	23.020	0.087
5	2	1	0.25	0.10	0.050	4.016	4.015	0.025
5	2	1	0.30	0.30	0.300	9.024	9.022	0.022
5	2	1	0.40	0.40	0.400	11.028	11.026	0.018
5	2	1	0.50	0.20	0.100	5.021	5.021	0.000
5	2	1	0.50	0.50	0.500	13.033	13.030	0.023
5	2	1	1.00	0.40	0.200	7.033	7.032	0.014
5	2	1	1.00	1.00	1.000	23.056	23.051	0.022
5	2	1	5.00	2.00	1.000	23.124	23.122	0.009

Table (4.12), shows the results for a set of problems where arrival and service rates and cost coefficients are as indicated. Here the values $\%^{SD}$ range from 0 to 0.087. These values are comparable to those of Tables (4.4) - (4.7) and it is clear that the index policies are again performing well.

Table 4.13: Results for system with $\lambda_1 = 1.101$, $\lambda_2 = 0.890$, $\lambda_3 = 0.719$, $\mu_1 = 14.128$, $\mu_2 = 5.803$ and $\mu_3 = 1.386$

b_1	b_2	b_3	c_1	c_2	c_3	C^{IND}	C^{SD}	$\%^{SD}$
5	2	1	0.00	0.00	0.000	3.030	3.030	0.000
5	2	1	0.01	0.02	0.050	3.630	3.576	1.510
5	2	1	0.02	0.04	0.100	4.230	4.123	2.595
5	2	1	0.05	0.02	0.010	3.159	3.156	0.095
5	2	1	0.05	0.10	0.250	6.030	5.763	4.633
5	2	1	0.10	0.10	0.100	4.256	4.202	1.285
5	2	1	0.10	0.20	0.500	9.002	8.496	5.956
5	2	1	0.20	0.20	0.200	5.482	5.374	2.010
5	2	1	0.20	0.40	1.000	14.780	13.962	5.859
5	2	1	0.25	0.10	0.050	3.677	3.663	0.382
5	2	1	0.30	0.30	0.300	6.707	6.546	2.460
5	2	1	0.40	0.40	0.400	7.930	7.718	2.747
5	2	1	0.50	0.20	0.100	4.325	4.297	0.652
5	2	1	0.50	0.50	0.500	9.150	8.890	2.925
5	2	1	1.00	0.40	0.200	5.620	5.565	0.988
5	2	1	1.00	1.00	1.000	15.213	14.751	3.132
5	2	1	5.00	2.00	1.000	15.957	15.705	1.605

Table (4.13), shows the results for a set of problems where arrival and service rates and cost coefficients are as in Table (4.12). Here the values $\%^{SD}$ range from 0.000 to 5.956. These values are again comparable to those of Tables (4.4) to (4.7) and it is clear that the index policies are performing at close to optimal.

4.10 Conclusions and future work.

The Whittle index policies derived in Chapter 3 perform at close to optimal levels for all of the problems considered. The problems all concerned single server M/M/1 queueing systems. The use of the achievable region as an effective means of assessing performance via the development of a lower bound on achievable cost would seem to be justified. This is likely to be of increasing importance in future work on larger, more complex problems where the computation of C^{OPT} is no longer possible. Such problems could perhaps involve cost rates $C(n)$ of order p where p is greater than 2 and where we know that the Whittle index to be a polynomial of degree $p-1$. We could also study performance in problems where there is a larger number of customer types. Extension of the work to more complex systems such as M/G/1 and multi-server systems are also future objectives.

Bibliography

Ansell, P. S., Glazebrook, K. D. & Mitrani, I. (2001), 'Threshold policies for a single server queuing network.', *Prob. Eng. Inf. Sci.* **15**, 15–33.

Ansell, P. S., Glazebrook, K. D., Mitrani, I. & Niño-Mora, J. (1999), 'A semidefinite programming approach to the optimal control of a single server queuing system with imposed second moment constraints.', *J. Oper. Res. Soc.* **50**, 765–773.

Ansell, P. S., Glazebrook, K. D., Niño-Mora, J. & O'Keeffe, M. (2003), 'Whittle's index policy for a multi-class queueing system with convex holding costs', *Math Meth Oper Res* **57**(1), 21–39.

Bellman, R. E. (1957), *Dynamic Programming*, Princeton University Press, Princeton, N,J.

Bertsimas, D. & Niño-Mora, J. (1996), 'Conservation laws, extended polymatroids and multi-armed bandit problems: a polyhedral approach to indexable systems.', *Math. Oper. Res.*

Bertsimas, D. & Niño-Mora, J. (2000), 'Restless bandits, linear programming relaxations and a primal-dual index heuristic.', *Operations Research* **48**(1), 80–90.

Bertsimas, D., Paschalidis, I. & Tsitsiklis, J. (1994), 'Optimization of multiclass queueing networks: polyhedral and nonlinear characterizations of achievable performance.', *Ann. Appl, Prob.* **1**, 43–75.

Blackwell, D. (1965), 'Discounted dynamic programming', *Ann. Math. Statist.* **36**, 226–235.

Blanc, J. P. C. (1993), Performance analysis and optimization with the power-series algorithm, *in* 'Performance Evaluation of Computer and Communications Systems', Vol. 729 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 53–80.

Cobham, A. (1954), 'Priority assignment in waiting line problems', *Oper. Res.*

Coffman, E. & Mitrani, I. (1980), 'A characterization of waiting time performance realizable by single server queues.', *Opns. Res.* **28**, 810–821.

Cox, D. R. & Smith, W. L. (1961), *Queues*, Metheun, London.

Federgruen, A. & Groenevelt, H. (1988), 'Characterization and optimization of achievable performance in general queueing systems.', *Operations Research* **36**, 733–741.

Gelenbe, E. & Mitrani, I. (1980), *Analysis and synthesis of computer systems*, Academic Press, London.

Gittins, J. C. (1979), 'Bandit Processes and Dynamic Allocation Indices', *J. R. Statist. Soc. B* **41**(2), 148–177.

Gittins, J. C. & Jones, D. M. (1974), A dynamic allocation index for the sequential design of experiments., *in* 'Progress in statistics (European Meeting of Statisticians, Budapest, 1972)', Vol. 9, Colloq. Math. Soc., János Bolyais, North-Holland, Amsterdam., pp. 241–266.

Glazebrook, K. D. (1976), 'Stochastic scheduling with order constraints', *Int. J. Sys. Sci.* **7**, 657–666.

Glazebrook, K. D. (1982), 'On sufficient conditions for superprocesses due to Whittle', *J. Appl. Prob.* **19**, 99–110.

Glazebrook, K. D. (1991), 'Competing Markov decision processes', *Ann. Oper. Res.* **29**, 537–564.

Harrison, J. M. (1975), 'Dynamic scheduling of a multiclass queue: discount optimality.', *Oper. Res.* **23**, 270–282.

Hooghiemstra, G., Keane, M. & van de Ree, S. (1988), 'Power series for stationary distributions of couple processor models', *SIAM Journal of Applied Mathematics*.

Klimov, G. P. (1974), 'Time sharing service systems I.', *Theory Prob. Appl.* **19**, 532–551.

Kojima, M. (1994), A primitive interior point algorithm for semidefinite programs in mathematica., Technical Report Research Report 293, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo Institute of Technology, Oh-Okayama, Meguro, Tokyo 152, Japan.

Nash, P. (1980), 'A Generalized Bandit Problem', *J. R. Statist. Soc. B* **42**(2), 165–169.

Niño-Mora, J. (2001*a*), Countable partial conservation laws, Whittle's restless bandit index and a dynamic $c\mu$ rule for scheduling a multiclass M/M/1 queue with convex holding costs., Technical report, Universitat Pompeu Fabra.

Niño-Mora, J. (2001*b*), 'Restless bandits, partial conservation laws and indexability.', *Adv. Appl. Prob.* **33**(1), 77–89.

Puterman, M. (1994), *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley, New York.

Shanthikumar, J. G. & Yao, D. D. (1992), 'Multiclass queueing systems: polymatroidal structure and optimal scheduling control.', *Operations Research* **40**, S293–S299.

Tcha, D. & Pliska, S. (1977), 'Optimal control of single-server queueing networks and multi-class m/g/1 queues with feedback', *Oper. Res.* **25**, 248–258.

Tijms, H. (1994), *Stochastic Models, an Algorithmic Approach*, Wiley, New York.

Van Meighem, J. A. (1995), 'Dynamic scheduling with convex delay costs: the generalised $c\mu$ rule', *Ann. Appl. Prob.* **5**, 809–833.

Vandenberghe, L. & Boyd, S. (1996), 'Semidefinite programming', *SIAM REVIEW* **38**, 45–95.

Varaiya, P., Walrand, J. & Buyukkoc, C. (1985), 'Extensions of the multi-armed bandit problem: the discounted case', *IEEE Trans. AC* **30**, 426–439.

Veatch, M. & Wein, L. M. (1996), 'Scheduling a make-to-stock queue: Index policies and hedging points.', *Oper. Res.* **44**, 634–647.

Weber, R. R. (1988), *Stochastic differential systems, stochastic control theory and applications.*, Springer-Verlag., chapter Stochastic scheduling on parallel processors and minimization of concave functions of completion times.

Weber, R. R. & Weiss, G. (1990), 'On an index policy for restless bandits.', *J. Appl. Prob.* **27**, 637–648.

Weber, R. R. & Weiss, G. (1991), 'Addendum to "On an index policy for restless bandits".', *Adv. Appl. Prob.* **23**, 429–430.

Weiss, G. (1988), 'Branching bandit processes.', *Prob. Eng. Inf. Sci.* **2**, 269–278.

Whittle, P. (1980), 'Multi-armed Bandits and the Gittins Index', *J. R. Statist. Soc. B* **42**(2), 143–149.

Whittle, P. (1981), 'Arm acquiring bandits', *Ann. of Prob.* **9**(2), 284–292.

Whittle, P. (1988), 'Restless Bandits: Activity Allocation in a Changing World', *J. Appl. Probab.* **25A**, 301–313.