On Optimal Policies for Energy-Aware Servers

ON OPTIMAL POLICIES FOR ENERGY-AWARE SERVERS

BY

VINCENT MACCIO, B.Eng.

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTING AND SOFTWARE

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

Master of Applied Science

Master of Applied Science (2013) McMaster University

(Software Engineering) Hamilton, Ontario, Canada

TITLE:	On Optimal Policies for Energy-Aware Servers
AUTHOR:	Vincent Maccio B.Eng., (Software Engineering and Management) McMaster University, Hamilton, Ontario, Canada
SUPERVISOR:	Dr. Douglas G. Down
NUMBER OF PAGES:	x, 127

I would like to dedicate this work to my loving wife, Stephanie Maccio. Her constant support (both emotionally and financially) has made it possible for me to be a successful Master's student. Her selfless encouragement and positive attitude has made me, and continues to make me, truly happy.

Abstract

As energy costs and energy used by server farms increase, so does the desire to implement energy-aware policies. Although under some cost functions, optimal policies for single as well as multiple server systems are known, large gaps in theoretical knowledge are present in the field. Specifically, there exists many widely used and non-trivial cost functions, where the corresponding optimal policy remains unknown. This work presents and leverages a model which allows for an exact analysis of these optimal policies with considerable generality, for on/off single server systems under a broad range of cost functions that are based on expected response time, energy usage, and switching costs. Furthermore, from the results derived in the analysis, several applications and implications are presented and discussed. This includes the determination of routing probabilities to show a range of non-trivial optimal routing probabilities and server configurations when energy concerns are a factor.

Acknowledgements

This work has only been made possible through the help of my supervisor Dr. Douglas G. Down. His patience and enthusiasm allowed me to learn and understand both the rudimentary and advanced topics of stochastic modelling and queueing theory. His guidance and insight allowed for my research to be fruitful and rewarding, while his friendly and good character allows me to look back on my time as a Master's student with only fond memories. All of which, I am eternally thankful for.

This research was funded by the Natural Sciences and Engineering Research Council of Canada.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The relative as well as absolute energy consumed by servers have been steadily increasing in North America over the past several years [4, 19]. As systems grow and expand, energy concerns have become a major factor for server farm managers from both environmental and economic viewpoints. However, the task of creating feasible optimal or near-optimal policies is a daunting problem due to the sheer complexity these systems exhibit. Even for single server systems, when energy is a factor, optimal policies remain unknown for a number of metrics considered in the literature. This work focuses on analysing energy-aware single server systems, with the prospect of deriving optimal policies.

When determining an optimal policy, one minimizes some cost function (possibly subject to some constraints). The cost function is constructed from system metrics which are desirable to keep low. For example, the expected number of jobs in the system, the expected response time of a given job, the expected energy used by the system, and the expected rate at which the server turns on and off. In Chapter 4, a model is

presented and discussed, which allows one to describe a set of optimal policies which the optimal policy is a member of. Furthermore, this is true for a large range of cost functions due to some convenient observed properties which these policies exhibit.

With the problem formulated and the model defined, Chapter 5 gives a detailed analysis under varying assumptions about the system. Specifically this chapter derives closed form expressions for the expected response time of a job, the expected energy used by the system, and the expected rate at which the server turns off, denoted $\mathbb{E}[R]$, $\mathbb{E}[E]$, and $\mathbb{E}[Sw]$ respectively. Furthermore, this is done with imposing little on the system in terms of assumptions, by allowing most underlying distributions to be considered in the general case. Here the impact that different system configurations have on these metrics is also examined in some detail.

After the model has been analysed, and the metrics solved for, Chapter 6 shows how the optimal policy is derived given particular cost functions. Furthermore, this chapter applies the model in different contexts of interest. Specifically, this chapter considers constrained optimization, the addition of sleep states, and invokes the results from Chapter 5 in the determination of routing probabilities in a multi-server system.

Chapter 2

Preliminary Knowledge

This chapter presents an explanation of some of the fundamental tools and concepts which appear in stochastic modelling and queueing theory. If the reader is familiar with Continuous Time Markov Chains, and preliminary models from queueing theory, this chapter may be skipped.

2.1 Stochastic Processes

A stochastic or random process is a mathematical abstraction used to represent and model a system's behaviour over time. Formally, a stochastic process is a set of random variables $\{X_t \mid t \in T\}$. The index set T is typically interpreted as a set of time values. The random variables X_i denote information concerning the system in question and may be either discrete or continuous, i.e. X_i may be the number of customers in a system (discrete), or the power being consumed by a system (continuous), at time i. In contrast, T may be countable, i.e. $T = \{1, 2, 3... \}$ or defined on some interval, i.e. $T = \{t \mid t > 0\}$.

2.1.1 Markov Processes

A stochastic process is a Markov process if the Markov property holds. The Markov property states that for every sequence of increasing time values $(t_0, t_1, t_2, ..., t_n)$, given the values of $X_{t_0}, X_{t_1}, X_{t_2}, ..., X_{t_{n-1}}$, the conditional distribution of X_{t_n} depends only on $X_{t_{n-1}}$. This is seen formally as,

$$\begin{aligned} &P[X_{t_n} \leq x_n | X_{t_0} = x_0, X_{t_1} = x_1, X_{t_2} = x_2, ..., X_{t_{n-1}} = x_{n-1}] \\ &\quad = P[X_{t_n} \leq x_n | X_{t_{n-1}} = x_{n-1}]. \end{aligned}$$

This has the interpretation that the stochastic process is "memoryless", that is the system's future behaviour only depends on its present state, and is completely independent from its past behaviour. Exploiting the memoryless property of a Markov process allows for an elegant analysis of such a system, since one can analyse all future behaviours of a system with only the knowledge of the current system state. This result is the cornerstone for the analysis of many queueing systems, which will be seen in Section 2.2.

2.1.2 Continuous-Time Markov Chains

A continuous-time Markov chain (CTMC) is a Markov process where the random variables $\{X_t \mid t \in T\}$ take on discrete values from some set S, called the state space, and the set T is defined to be some continuous interval. A CTMC is often thought of as a directed graph where the nodes of the graph are the elements of S (the system states), and the arrows are the "transition rates" between states, labelled by $q_{i,j}$, the rate the system moves from state i to state j. Given all of the transition rates, one

can construct the transition matrix for a given Markov chain as shown in (2.1),

$$Q = \begin{pmatrix} -q_{0,0} & q_{0,1} & q_{0,2} & \cdots \\ q_{1,0} & -q_{1,1} & q_{1,2} & \cdots \\ q_{2,0} & q_{2,1} & -q_{2,2} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{2.1}$$

where $q_{i,i} = \sum_{j \neq i} q_{i,j}$. This last relation comes from the fact that for each state, the sum of probabilities to move to any other state (including the given state) in a given amount of time, equals 1. Table 2.1 shows the different classes of stochastic processes when switching between discrete and continuous state spaces.

	Time Values	
State Space	Discrete	Continuous
Discrete	Discrete-Time Markov Chain	Continuous-Time Markov Chain
Continuous	Discrete-Time Markov Process	Continuous-Time Markov Process

Table 2.1: Stochastic Process Classes

There is a special class of CTMCs called birth-death processes. This is when S is isomorphic to some subset of the natural numbers, where for simplicity it is often assumed that $S \subseteq \mathbb{N}$. It is also the case that the state variable (current state) may only increase or decrease by a value of 1 between each transition. The transition matrix for birth-death processes is therefore of the form,

$$Q = \begin{pmatrix} -\lambda_0 & \lambda_0 & 0 & \cdots \\ \mu_1 & -(\lambda_1 + \mu_1) & \lambda_1 & \cdots \\ 0 & \mu_2 & -(\lambda_2 + \mu_2) & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

and may be finite or infinite. In this context, λ_n and μ_n are referred to as the birth and death rates in state n, respectively. Birth-death processes arise in many different fields of study such as biology, demography, and engineering, however the main focus here is applying them to different queueing networks. Although simplistic, many rudimentary queuing systems become birth-death processes once exponential assumptions on the underlying distributions are imposed.

2.2 Queueing Theory

Queueing theory is the mathematical study and analysis of "lines" or "queues" where the goal is to make statistical predictions on the characteristics of systems where these queues are present. In general these systems have arrivals of customers or jobs which occur according to some random process. These jobs are sent to queues where they wait to be served or processed. Servicing a job also takes a random amount of time following some distribution. These systems can become extremely complex by connecting multiple systems together, implementing different routing policies, having different arrival streams, etc. making for interesting and challenging problems. This chapter presents some of the simpler queueing models, and the methods used to analyse them.

2.2.1 Kendall's Notation

When the systems in question are limited to having a single queue to which all jobs arrive, there still exists many different parameters in order to describe the system or queue fully. Due to the complexity and variation in behaviour these queues exhibit,

a convenient notation known as Kendall's Notation is widely used. The notation is a list of six parameters delimited by "/" of the form $A/S/c/K/N/D$. A denotes the distribution of the times between arrivals, S denotes the distribution of the services times of the jobs, c denotes the number of servers, K denotes the capacity of the queue, N denotes the population sized to be served, and D denotes the order that the jobs get served when waiting in the queue. While c, K, and N are all natural numbers, there exists a further notation denoting the type of distributions A and S follow. The notation for the distributions mentioned in this work is presented in Table 2.2 (the list is only partial). Different instantiated values for D are given in Table 2.3 but for all models presented in this paper the FIFO policy is used.

Notation	**Distribution**	**Description**
M	Exponential	Standing for Markovian or memoryless, this distribution is often used to allow for an analysis using CTMCs. In general, this imposition on the distribution is restrictive.
E_k	Erlang k	An Erlang distribution with shape parameter k still allows for easy analysis since it is composed of k exponential distributions in series while still giving flexibility when fitting the model to observations.
D	Degenerate	The distribution with 0 variance, that is the random variable which follows this distribution will always equal the mean. This is used when times between arrivals or service times are constant.
G or GI	General Independent	Represents the possibility of any distribution, allowing for a completely general analysis of the system.

Table 2.2: Distribution Notation

It is common when denoting these queues to drop the last three parameters and simply write $A/S/c$. When this is done, the following values for the excluded parameters

Shorthand	Name	Description
$FIFO$	First In First Out	Perhaps the most intuitive policy, the jobs are processed in the order they arrive.
$LIFO$	Last In First Out	Jobs are processed in the reverse order they arrive. The policy must also make the choice if a job is preempted if a new job arrives while it is being processed.
PS	Processor Sharing	Each job in the system gets an equal fraction of the processor.

Table 2.3: Queue Policies

are assumed to be $K = \infty$, $N = \infty$, and $D = FIFO$. For example an $M/M/2$ queue is a system where interarrival as well as service times are exponentially distributed, there are two servers, the buffer (or queue) has no maximum capacity and implements a First In First Out policy, and the number of jobs available to arrive to the system is infinite.

When analysing these systems mathematically, it is standard to denote the arrival rate as λ, the service rate from each server as μ, and the system utilization as ρ. It follows from here that $\rho = \lambda/c\mu$. This is also the condition for the system's stability. The system is considered unstable if jobs arrive to the system faster than they can be processed i.e. $\lambda \geq c\mu$. This implies that in a stable system $\rho < 1$. When analysing any of these systems this condition arises in the analysis, as will be seen in the remaining subsections in this chapter as well as Chapter 5.

2.2.2 Little's Law

Before any analysis of these systems is presented, it is important for the reader to be made aware of and understand a result in queueing theory known as Little's Law

[14]. It is one of the fundamental theorems in the field which allows one to derive expressions which would be hard to otherwise attain. The law states that the long term mean number of customers or jobs in a stable system is equal to the product of the arrival rate to the system and the mean time a job spends in the system. This law is applied to most queueing systems as the expected number of jobs in the system equals the arrival rate multiplied with the expected response time of any given job.

$$\mathbb{E}[N] = \lambda \mathbb{E}[R]$$

This law holds independently of the arrival distribution, the number of servers in the system, the type of policies the system implements, etc. It is regarded as one of the most important results in queueing theory, and is invoked many times in this work.

2.2.3 The M/M/1 Queue

In queueing theory, one of the most basic and easy to analyse systems is the M/M/1 queue. From the notation introduced in Section 2.2.1, this is a single server system where the interarrival times, as well as the service times of the jobs are both exponentially distributed. The interarrival times being exponentially distributed is equivalent to the arrivals following a Poisson process and is often referred to as such. Due to the exponential distributions, the memoryless property allows the system to be be modelled as a CTMC, where the state denotes the number of jobs in the system (waiting in the queue, and being processed). This Markov chain is depicted in Figure 2.1. Furthermore, since the state of the CTMC can only increase or decrease by at most one in any given transition, the system is also a birth-death process, where the transition matrix is given in (2.2).

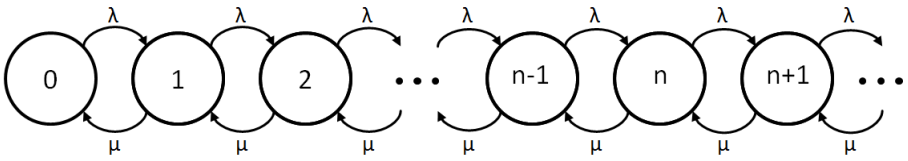


Figure 2.1: $M/M/1$ Queue

$$Q = \begin{pmatrix} -\lambda & \lambda & 0 & \cdots \\ \mu & -(\lambda+\mu) & \lambda & \cdots \\ 0 & \mu & -(\lambda+\mu) & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{2.2}$$

It is of interest to capture the system's behaviour in steady state. This means allowing time to approach infinity and observing the proportion of time the system spends in each of its states. This steady state proportion of time is often referred to as the system's steady state distribution and for each state, n, its value is denoted by π_n. The reason why this type of analysis is convenient is that it allows one to determine system metrics such as the distribution of the number of jobs in the system. From here, the expected number of jobs in the system can be obtained, and by applying Little's Law, one can arrive at the expected response time for any given job.

It is known that the rate into each state must equal the rate out of that state. This is true because the number of times the system enters the state, and the number of times which the system leaves that same state, may differ by at most one. Letting time go to infinity it follows that the rate in and out of the state must be equal. From this observation, for any state $n > 0$ it is seen that $\lambda\pi_{n-1} + \mu\pi_{n+1} = (\lambda+\mu)\pi_n$ (rate in equals rate out). Iterating over all $n > 0$ gives us a set of equations referred to as

the balance equations for the CTMC shown in Figure 2.1. We also have the balance equation for state 0, $\lambda\pi_0 = \mu\pi_1$. Rearranging gives us $\pi_1 = \rho\pi_0$ and from a simple recursion we find $\pi_n = \rho^n\pi_0$. This gives a set of equations with an infinite number of solutions, so some sort of boundary equation must be invoked. It is noted that the sum of all of the steady state probabilities must equal 1, in other words at any point in time the system must be in exactly one of its states. This is seen mathematically as,

$$\sum_{i=0}^{\infty} \pi_i = 1 \quad \Rightarrow \quad \pi_0 \sum_{i=0}^{\infty} \rho^i \quad \Rightarrow \quad \pi_0 = 1 - \rho.$$

Putting the boundary and balance equations together, the steady state distribution of the number of jobs in the system is given by

$$\pi_n = (1 - \rho)\rho^n.$$

From here, we can weight each π_n by n and sum them to arrive at the expected number of jobs in the system, and with an application of Little's Law, the expected response time. The algebra will not be shown here as the results are well known. These metrics are given by

$$\mathbb{E}[N] = \frac{\lambda}{\mu - \lambda} \quad \text{and} \quad \mathbb{E}[R] = \frac{1}{\mu - \lambda}.$$

For the purposes of this work, the energy used by these systems is also of interest. Although usually not mentioned in the literature, the expected energy used by an M/M/1 queue is easily determined. Assuming there is an amount of energy used while

processing jobs and when the system is idle, denoted by E_{Busy} and E_{Idle}, respectively, the expected energy used by the system ($\mathbb{E}[E]$) is simply solved by weighting the probabilities of being busy or idle by the energy values. The probability of being idle is π_0, and the probability of being busy is $1 - \pi_0$, (the utilization ρ). Putting it together, the expected energy is given by

$$\mathbb{E}[E] = E_{Busy}\rho + E_{Idle}(1 - \rho).$$

The exponential assumptions make for a system that is clean and easy to analyse, however this limits the feasible application of the model in practical settings. In the next section we relax some of these assumptions to make for a more general model, but also one that is more challenging to analyse.

2.2.4 The M/G/1 Queue

One of the more interesting and practical queueing models is the $M/G/1$ queue. This is due to the nature of the assumption that the arrival stream being a Poisson process is reasonable in many applications, i.e. server requests being generated all over the country. On the other hand, exponentially distributed service times, ($M/M/1$) generally is a poor assumption. Furthermore, exact analytic results for the $M/G/1$ are well known and relatively simple to apply, if the first and second moments of the service time distribution is known.

The difficulty in analysing such a system is the memoryless property is not present in all underlying distributions, specifically it is not a property of the service time distribution. Without the memoryless property, the system cannot be analysed as a

Markov chain where the states are the number of jobs in the system, since one would also have to keep track of how long a current job has been processed to make predictions about the future state. To overcome these challenges, the system is inspected every time a job leaves the system. At this exact point in time all that is needed to be known to predict future events is the current number of jobs in the system. This is due to the memoryless nature of the arrival stream as well as the knowledge that the current job has not commenced processing. This is an example of an embedded Markov chain. Analysing this embedded Markov chain allows one to give an exact analysis of the system, however before the details are presented, some notation must be introduced. Let N_n be a random variable denoting the number of jobs in the system at the n^{th} departure point (the moment the n^{th} job leaves the system). Let A_{n+1} be a random variable denoting the number of jobs which arrive to the system during the service time between the n^{th} and $(n+1)^{th}$ departure points. This gives the following recursion,

$$N_{n+1} = \begin{cases} N_n + A_{n+1} - 1 & N_n \geq 1 \\ A_{n+1} & N_n = 0 \end{cases}.$$

Noting that all service times are identically and independently distributed (i.i.d), the index on A is dropped and referred to from this point on as A_s. Using the Heaviside function this can be rewritten as,

$$N_{n+1} = N_n - \mathcal{U}(N_n) + A_s. \tag{2.3}$$

The goal is to determine the expected number in the system in steady state, this is

equivalent to letting $n \to \infty$, and taking expectations of both sides of (2.3), however if this were to be done the expected number in the system would cancel out and there would be no hope of solving for it. To work around this issue, both sides of (2.3) are squared before expectations are taken. This yields the equation

$$0 = \mathbb{E}[A_s^2] + 2\mathbb{E}[N]\mathbb{E}[A_s] - \mathbb{E}[\mathcal{U}(N)] - 2\mathbb{E}[N] - 2\mathbb{E}[\mathcal{U}(N)]\mathbb{E}[A_s].$$

After some calculations, and letting σ_S^2 denote the variance of the service time distribution, one can solve for $\mathbb{E}[N]$ and with Little's Law $\mathbb{E}[R]$.

$$\mathbb{E}[N] = \rho + \frac{\rho^2 + \lambda^2\sigma_S^2}{2(1-\rho)} \quad \text{and} \quad \mathbb{E}[R] = \frac{1}{\mu} + \frac{\rho^2 + \lambda^2\sigma_S^2}{2\lambda(1-\rho)}$$

Again, for the purposes of this work the expected energy consumed by the system will be required. For all single server systems, the utilization is known to be ρ, and due to this the expected energy of an $M/G/1$ queue is equal to that of an $M/M/1$. From the previous section, this is known to be

$$E_{Busy}\rho + E_{Idle}(1-\rho).$$

This shows the expected energy of the system to be completely independent of the underlying service time distribution. This same result is presented in the analysis of energy-aware systems in Chapter 5.

	Metric		
Queue	$\mathbb{E}[N]$	$\mathbb{E}[R]$	$\mathbb{E}[E]$
M/M/1	$\frac{\lambda}{\mu - \lambda}$	$\frac{1}{\mu - \lambda}$	$E_{Busy}\rho + E_{Idle}(1-\rho)$
M/G/1	$\rho + \frac{\rho^2 + \lambda^2\sigma_S^2}{2(1-\rho)}$	$\frac{1}{\mu} + \frac{\rho^2 + \lambda^2\sigma_S^2}{2\lambda(1-\rho)}$	$E_{Busy}\rho + E_{Idle}(1-\rho)$
G/G/1	Not known	Not known	$E_{Busy}\rho + E_{Idle}(1-\rho)$

Table 2.4: Queueing Theory Equations

2.2.5 Queueing Equations

The M/M/1 and M/G/1 queues are some of the most widely understood and used systems in queueing theory. Knowledge of both of these systems is assumed in this work and for convenience these equations are collected in Table 2.4. While this chapter provided the necessary basics to understand the mathematics and reasoning in the later parts of this work, there were some subtleties and details which were omitted. If the reader is looking for deeper understanding of the ideas presented here, or perhaps a broader look at queueing theory in general the following literature is recommended; [10, 11, 12]. Furthermore, once the reader grasps the basics of the field and wishes to gain even deeper insight into the analytical methods used in the previously cited works, [17] is also recommended.

Chapter 3

Literature Review

This chapter examines other articles and contributions which pertain to the field to which this work belongs. In addition, this chapter discusses the contributions of this work to the field.

3.1 Green Computing

Green or sustainable computing is a relatively new field of research in which the trade-offs of performance and energy or power usage of computing systems are examined and analysed. While having these systems always operating at their top performance is an inarguably effective policy to implement, it may not be the most efficient use of the systems resources. This could lead to higher system costs than necessary (paying for power while a server idles). Also there may exist the benefit of being (or being perceived to be) green. Different authors have different motivations for the research, but the basic idea of looking at how these different system metrics interact as configurations change, remains the same.

While the motivation for solving these problems is intuitive or intrinsic, several data center statistics or facts are given in [7, 13, 16, 22]. These works show that server farms use a relatively large portion of total energy used across North America, and that both the absolute and relative values of their energy use are on the increase. Furthermore, the article [3] gives a more thorough argument why these energy concerns are valid. Finally, [2] takes a numerical viewpoint, and shows that in data centers, a typical server will idle a non-trivial proportion of time, and use roughly between 60%-70% of the energy that it would use while processing jobs during that time.

With the inherent application to industry, the field has many active researchers. These problems can vary from inspecting an isolated single server system, to looking at the whole computational framework of data centers across the continent. However, although the specific problems vary across works, the nature of the field always has authors looking at the trade-offs of performance and resource consumption. Due to this, authors will typically look to do their analysis with respect to some cost function. In the literature, the research can be segregated into using one of two types of cost functions. The cost function used by [6, 7, 9] is $\mathbb{E}[R]\mathbb{E}[E]$, while [3, 15, 16, 18, 20, 22, 23] used a cost function of the form $\mathbb{E}[R] + \beta_1\mathbb{E}[E] + \beta_2\mathbb{E}[Sw]$. While both cost functions have their advantages and disadvantages, analysing systems and policies under only one or a small set of cost functions can be problematic. The reader will see that this is one of the main issues addressed in this thesis.

Many articles deal with the same type of problems which are analysed in this thesis, but approach it from different directions. For example, the works of [3, 16, 22] looked at determining the optimal configuration of a server farm when the job sizes are known at arrival, and the decision to turn servers off or keep them on is made at discrete time intervals. This is then formulated as an optimization problem and solved for. The article [3] was the first to appear and accounts for wear and tear cost on the servers by allowing for an $\mathbb{E}[Sw]$ term in the cost function. The article [15] added the variation that jobs can be routed to different geographical locations where energy costs may differ. The work in [16] took a different viewpoint where different customers pay a certain amount, based on a function of the response time of that job. The work of [18] looked at a similar problem where jobs are routed to separate on/off queues, and the problem was solved using Markov Decision Processes (MDPs).

The work of [13] took a different approach to the same style of problem where the decision of when and what servers to turn on and off is made with the goal of minimizing the mean response time. Furthermore, the servers can be completely heterogeneous. However, the optimization problem is constrained by a maximum power value. This has the interpretation that the system has a limited energy supply and one wishes to use the power in the most efficient way possible. This paper stands out as in this setting the optimal policy is not the one which minimizes some constructed cost function, but rather one which minimizes $\mathbb{E}[R]$ under energy constraints. In the unconstrained case, the optimal policy is one which simply minimizes some given cost function, while the way the problem is presented here, one can interpret the optimal policy as maximizing efficiency of the system given a constant power supply.

The articles [22, 23] explored the issue of speed scaling in the context of green computing. Speed scaling refers to the ability to put more power into a server causing the processing rate to increase. This of course is another trade-off option between performance and energy. The authors examined these systems and provide both a stochastic and worst case analysis, where the job sizes are known upon arrival. The primary focus is on making two decisions. Firstly, which scheduling policy should the system employ (FIFO, PS etc.). Secondly, how much power should be put into the system at a given time. The authors are able to provide many insights. However, the policies examined are found to be only near-optimal, and many of the insights are seen numerically.

3.2 Vacation Models

Within the study of classic stochastic models ($M/M/1$, $M/G/1$, etc.), there are a subset of models called *vacation* models. This refers to systems where the server can be in a state where it cannot process jobs. When the server is in such a state it is referred to as being on vacation. These models are of great interest as they have many natural applications to different areas of industry, such as manufacturing, telecommunications, healthcare, and customer service to name a few. The primary motivation for understanding, examining, and extending these models in this work, is that the *vacation time* of the server can be viewed as the time which the server is turned off.

Many authors have analysed different variations of these vacation models. The book [21] describes in detail many of these systems and the subtle differences between them. Perhaps the simplest vacation model is one in which the server begins its vacation period as soon as there are zero jobs in the system. The time the server spends in this vacation state is exponentially distributed with some rate, say v. If the server ends its vacation before a new job arrives to the system, it immediately begins a new vacation period. Due to this behaviour and the properties of the exponential distribution, this system is equivalent to a system which remains on vacation until a job arrives, at which time it will start a new vacation period (again with rate v) and proceed to process the newly arrived job once the vacation ends. In general, systems which begin their vacation as soon as the server idles are called *exhaustive*, and are denoted in a modified Kendall notation where the list of vacation parameters is presented as a list in () after the classical counterpart. For example, under exponential assumptions for all underlying distributions, the previous model is denoted as $M/M/1(E)$. The article [5] as well as [21] show that the number in the system, as well as the response time for such a system can be seen as a decomposition of two random variables, where one of the variables is the corresponding random variable for the classical counterpart. Furthermore, this decomposition property holds when the processing and vacation times follow general distributions.

While this decomposition result has been known for some time, even under exponential assumptions, the steady state distribution of the system, as well as closed form expressions for metrics such as the expected response time, remained unknown until it was recently solved in [8]. However, under general assumptions, closed form

expressions remained unknown, but are later presented in this work.

In Chapter 4, a model is presented which can be seen as a combination of several of the vacation models described in [21]. These different types of vacation models are as follows. Firstly, the model is *non-exhaustive*, meaning that the server may not vacation as soon as it idles. Secondly, the model is of the *setup* family. This means that when the system meets some condition to start ending its vacation, this may take some random amount of time to achieve. Lastly, the system follows a *threshold policy*, meaning that once a certain number of jobs arrive to the system while the server is on its vacation, the system proceeds to end the vacation period. Although all of these systems are analysed and discussed in [21], a system which incorporates all of the above behaviour has not yet been examined in other works.

Other authors have also researched these vacation models in the context of multi-server system. Certain expressions and properties can be derived from these models. However, due to the introduced complexity which adding multiple servers bring to the system, many assumptions must be imposed. For example, the article [24] analyses an $M/M/c$ system with an (e, d) policy, where one of the limitations of this system is the number of jobs present in the system when a new server starts to end its vacation is equal to the number of jobs in the system when a server should shut down. While such systems have their advantages, it is clear that they do not describe optimal configurations.

3.3 Previous Work

Optimal policies for on/off servers is an interesting and active topic. As such, other authors have previously done research which directly relates to some of the work presented here. Specifically, this thesis was highly influenced by the research done in [8, 6, 7, 9]. Here, different on/off server systems are modelled both in the single and multi-server settings. However, when attempting to establish optimal policies, the authors gave a perhaps narrow view of what an optimal policy is defined to be. Specifically, in these articles, the aim is to minimize a cost function which is the product of the expected energy used by the system and the expected response time of a given job, which they call the Energy Response Product (ERP). The following contributions of their work are all under the context of this specific cost function.

- For a single server system with sleep states, [7] defined the set of policies which the optimal policy must belong to. Furthermore, they showed a property of the ERP cost function, which will always be minimized if the server begins to turn on once a single job arrives to the system. However, here exponential assumptions on the underlying distributions were imposed.

- They derived the steady state distribution of an $M/M/c$ ($E\ SU$) (exhaustive and setup) system. This allows for the derivation of system metrics such as $\mathbb{E}[R]$, and gives further analysis for the optimal policies under the ERP cost function, due to its optimal properties discussed above.

- They examined and simulated several different intuitive policies in multi-server settings to give insight into how these systems behave. However, while this gives intuition for how these policies compare to each other, it is hard to see

how they compare to the optimal policies, as they remain unknown.

- In [6], a method termed the Renewal Reward Recursion (RRR) was described, which allows one to determine optimal policies for certain two dimensional Markov chains. Again, this allows for the exact analysis of simpler policies but the true optimal policies remain to be solved. Furthermore, exponential assumptions on the underlying distributions must be imposed.

As one can see, a considerable amount of work has already been done on this subject. However, some drawbacks certainly do exist. Firstly, the authors focused on only minimizing the ERP when determining optimal policies. As stated before, other authors favoured the cost function of $\mathbb{E}[E] + \mathbb{E}[R]$, for which optimal policies are not touched on in these works. Secondly, while they gave the exact analysis of many interesting and feasible policies, in the multi-server context, the optimal policies remain unknown. Lastly, much of their analysis assumed exponential distributions on the system, which could be unreasonable. While in some sections they did examine certain results under different distributions that are combinations of exponentials such as Erlang and hyper-exponential, they did not provide results for general distributions.

The article [1] also deserves mention here as the type of systems they analysed are similar to the systems which are presented in this thesis. The author looked at $M/G/1$ systems, which have vacations. The three different types of policies which the author considered are as follows. Firstly, an N type policy where the server turns back on once a certain number of jobs arrive to the system. Secondly, a D type policy where the server turns on once the expected time to process the jobs exceeds a certain value. Finally, a T type policy where the server turns back on after a certain threshold time

after its last busy period. The author was able to show that changing the cost function changes which of the three policies is optimal. While the paper is able to analyse the three polices under general service times, the model does not include setup times for the server. That is, when the server chooses to end its vacation, the server instantly turns on.

The authors of [20] looked at multi-server systems under a specific policy. They analysed these systems by modelling them as Markov chains (under full exponential assumptions) and then solved them numerically. Again, the set of policies which they described does not necessarily contain the optimal policy, even in the single server setting. The authors made conclusions based on a broad range of parameter values.

Although the field of green computing is relativity new, a lot of progress has already been made in understanding these systems. Many authors looks at these problems from many different angles, offering a broad viewpoint of energy trade-offs in data centers. However, due to this broad outlook, many gaps in the field exist. Specifically, optimal policies for data centers under general cost functions remain unknown. This thesis gives a model and analysis in this setting, which yields several results in the single server setting under general cost functions and distributional assumptions, allowing for some of these theoretical gaps to be filled in.

Chapter 4

Problem Formulation

Given a single server system where the server can be in one of two energy states, the following metrics are of interest: the expected response time of a job in the system, the expected number of jobs in the system, the expected energy consumed by the system, and the expected rate at which the server switches between energy states. From these metrics, one can construct a cost function associated with the system. It is desirable to derive a policy in which the cost function is minimized, primarily determining when the server should move between energy states. Even in the rudimentary single server setting, optimal policies for the majority of non-trivial cost function are unknown. Here it is shown how these systems are modelled, and how one can arrive at optimal policies from these formulations.

4.1 The Model

The system is modelled as having a high and a low state. The system may instantly move from the higher state to the lower, while it takes time to move from the lower

state to the higher. While in the lower state the system cannot process jobs. Given this, such a system can be further broken down to being in one of four energy states, *LOW, SETUP, BUSY, IDLE*. These energy states denote the current behaviour of the system from an energy stand point. Firstly, *LOW* corresponds to the system being in its lower state. Secondly, *SETUP* corresponds to the system currently transitioning from its lower to its higher state. Thirdly, *BUSY* corresponds to the server being in its higher state while processing jobs. Lastly, *IDLE* corresponds to the system being in its higher state, but not processing jobs. Each of these energy states also has an associated energy level, E_{Low}, E_{Setup}, E_{Busy}, and E_{Idle}, respectively. If $E_{Low} = 0$, the energy state *LOW* is renamed to *OFF*. For the majority of the analysis presented in Chapter 5, and for the remainder of the model description it is assumed $E_{Low} = 0$. However, the reader should remember that the analysis is robust enough to disregard this assumption, and in fact does so in Section 6.3. It is often more relevant when analysing optimal policies to know the ratio of the energy levels. We take these ratios with respect to E_{Busy}, and denote them as r_{Low}, r_{Setup}, and r_{Idle}. For ease of intuition and description for the remainder of this work, the lower energy state will often be referred to the server being off, the higher energy state as being on, and transitioning from the lower to higher state is referred to as the server turning on.

Jobs arrive to the system in a FIFO queue according to a Poisson process with known rate λ. When in state *OFF*, the system allows for k jobs to accumulate. Once the k^{th} job arrives, the server immediately begins to turn on; the system enters the state *SETUP*. The amount of time the system remains in *SETUP* is generally distributed with rate γ. Once the server turns on it begins to process the initial k jobs as well as

any jobs which arrived while it was turning on; the system moves to state *BUSY*. The server processes the jobs following some general distribution with rate μ. Once all jobs in the system are processed, the system enters the state *IDLE*. Here the system keeps track of the total time the system has been idling since the last time it turned on. The server will idle for an amount of time which is generally distributed with rate α before it moves to its lower energy state; the system enters state *OFF*. The amount of time that the server idles for before it turns off is referred to as the idle threshold. If a job arrives before the idle time reaches the idle threshold, the server begins to process it; the system enters state *BUSY*. From here the server will eventually become idle again, where the system once again can either move to *BUSY* or *OFF* depending on future events. This switching between *IDLE* and *BUSY* will continue until the total idling time of the server, since the last time it turned on, reaches the idle threshold (in which case the server turns off). From here, the behaviour repeats itself as if the server started in state *OFF* with no jobs in the system. It should be noted that if the idle threshold values follow the exponential distribution, each time the server becomes idle, the total time the system has spent idling since the last turn on time can be seen as being reset to 0 due to the memoryless property of the exponential distribution. This is a result which is exploited later in Chapter 5. The Table 4.5 summarizes the notation and parameters of the model.

4.2 Notation

To denote these systems, a composition of two sets of parameters is used i.e. $\{\} \circ \{\}$. The first set of parameters is given in classical Kendall notation to describe the non-energy-aware portions of the system. The set of parameters listed after the

Parameter(s)	Explanation
$E_{Low}, E_{Setup}, E_{Busy}, E_{Idle}$	The energy values associated with the different system states.
r_{Low}, r_{Setup}, r_{Idle}	The ratios between the energy values and E_{Busy}.
λ	The arrival rate of jobs to the system.
μ	The server's processing rate.
γ	The rate at which the server moves to its higher energy state from the lower.
α	The rate at which a server waits while idle before moving to its lower energy state.
k	The number of jobs the system allows to accumulate in the queue before beginning to move to the higher energy state.

Table 4.5: Parameter Summary

composition symbol are all parameters which are incorporated due to energy concerns. The first of these parameters is the turn on time distribution of the server, the second is the idling threshold distribution, and the last is the number of jobs allowed to accumulate before the server begins to turn on. For example, a queue with exponential assumptions on all four distributions that begins to turn on once k jobs arrive is an $M/M/1 \circ \{M, M, k\}$, system while if the job service times along with the server turn on times follow general distributions, the system would be an $M/G/1 \circ \{G, M, k\}$. The reason for denoting the systems in this way, as will be shown later, is that their metrics can be written as a decomposition where one of the terms will be the corresponding metric of the non-energy-aware counterpart (the first set of parameters). This would suggest these energy-aware queues themselves can be seen as a composition of the corresponding classical Kendall queues and the energy parameters.

4.3 Cost Functions

In order to determine the optimal policy, a cost function must be defined. Cost functions are built upon the system metrics of the expected number of jobs in the system, $\mathbb{E}[N]$, expected response time of a job, $\mathbb{E}[R]$, the expected energy used by the system, $\mathbb{E}[E]$, and the expected switching rate, or the expected rate which the server turns off, $\mathbb{E}[Sw]$. Using these metrics the following class of cost functions can be defined by:

$$f(\beta, w) = \sum_{i=0}^{M} \beta_i \mathbb{E}[N]^{w_{N,i}} \mathbb{E}[R]^{w_{R,i}} \mathbb{E}[E]^{w_{E,i}} \mathbb{E}[Sw]^{w_{Sw,i}}, \tag{4.4}$$

where β is a vector of weight values for each term and w is a matrix of the specific weights in the power of each metric contained in the weighted terms, and $\forall i.0 \leq \beta_i, w_{R,i}, w_{E,i}, w_{Sw,i}$ and are of the appropriate units. It is noted that due to Little's Law, the $\mathbb{E}[N]^{w_{N,i}}$ component of (4.4) can be removed, by adding the $w_{N,i}$ value to $w_{R,i}$ and by rescaling β_i by λ. This yields the same class of cost functions but gives a simpler form of

$$f(\beta, w) = \sum_{i=0}^{M} \beta_i \mathbb{E}[R]^{w_{R,i}} \mathbb{E}[E]^{w_{E,i}} \mathbb{E}[Sw]^{w_{Sw,i}}. \tag{4.5}$$

Again $\forall i.0 \leq \beta_i, w_{R,i}, w_{E,i}, w_{Sw,i}$ and are of the appropriate units.

4.3.1 Optimal Policies

Now that the family of cost functions has been defined, it must be shown that the model described in Section 4.1 can be leveraged to arrive at the optimal policy. To show this is equivalent to showing the model can describe the policy which minimizes

any of the cost functions contained within the class described in (4.5). The model makes two key assumptions about every policy it describes.

- The decision to start transitioning from the lower to the higher energy state (the decision to turn the server on) is made at the moment a job arrives to the system.
- If there are jobs in the system and the server is in its higher energy state, the server will never move to its lower energy state (the server will turn off only when it is idle).

If it can be shown that the policies which minimize the cost functions have these properties, then it can be inferred that the model can always describe the optimal policy. The first assumption is made without loss of generality due to the memoryless property of the arrival stream (the same decision would be made at any point in time between arrivals). The second assumption is a property of the optimal policy due to the nature of the cost function. If the system were to turn the server off while a job(s) remains in the system, $\mathbb{E}[R]$ will increase, since the job(s) that was in the system when it turned off must now wait until the system turns on before it can be completed. Furthermore, the $\mathbb{E}[Sw]$ component of the cost function would also increase since the server is turning off at a point where it could have remained on instead. At the same time, the system does not gain any benefit with respect to the $\mathbb{E}[E]$ component since it will still have to expend energy to complete the job(s) in the system at some point in the future. Due to the weights (β_i) being positive, i.e. it is never advantageous to increase $\mathbb{E}[R]$ or $\mathbb{E}[Sw]$ while holding all other values constant, it is concluded that the second assumption is a property of policies which

minimize the cost functions. This along with the first assumption being made without loss of generality, it follows that the optimal policy can be described by the model.

It is worthwhile to give another property of the optimal policy. Similar to the argument made to justify the servers beginning to turn on only when an arrival occurs to the system, the decision to turn a server off or keep it on is only made when a job departs the system and leaves it idle. This would imply that in the model, in any policy which minimizes the cost, $\alpha = 0$ or $\alpha \to \infty$. We leave α as part of the model for several reasons. Firstly, it gives insight on how scaling between these two extremes affects the system. Secondly, it allows one to easily determine where in the parameter space the optimal policy switches between $\alpha = 0$, and $\alpha \to \infty$. Thirdly, it allows for easier extensions of the model where this property may not necessarily hold. For example, this property does not hold when the arrivals do not follow a Poisson process, or in a multi-server setting. Lastly when optimizing under different conditions, i.e. minimizing a linear function of $\mathbb{E}[E]$ with a constraint on $\mathbb{E}[R]$, the optimal value of α could lie anywhere on the positive real line.

Chapter 5

Analysis

This chapter presents and analyses various instantiations of the model described in Chapter 4 by imposing assumptions on the underlying distributions as well as some of the turn on criteria (the value of k). The analysis starts with the assumption that all underlying distributions are exponential. Later sections progressively relax these assumptions and build to a general analysis allowing one to derive some practically useful results.

5.1 The M/M/1 $\circ$ {M,M,1} Queue

The simplest non-trivial instantiation of the presented model is imposing exponential assumptions on all underlying distributions, as well as limiting the number of jobs to build up in the system before moving to *SETUP* to be held constant at one ($k = 1$). While the assumptions on the distributions are not ideal due to potential modelling inaccuracies, the limitation on k is perhaps more detrimental when making design decisions based on the analysis. This is due to the fact that the optimal policy under

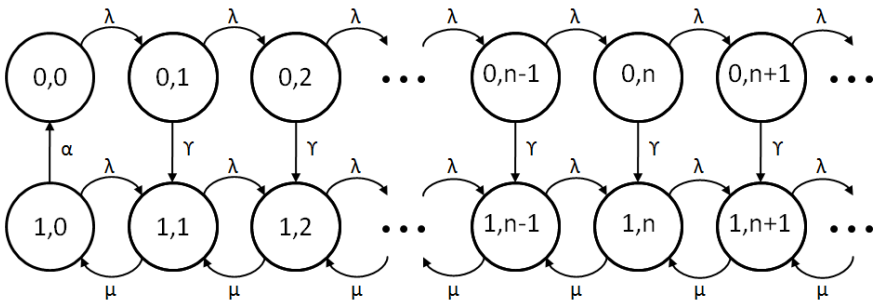


Figure 5.2: $M/M/1 \circ \{M, M, 1\}$ Queue

all cost functions of the form (4.5), cannot be described with an $M/M/1 \circ \{M, M, 1\}$ queue even if the exponential assumptions are justified. Specifically in the optimal policy $k > 0$ for some natural number. This leads to the question of, why is this system worth analysing? There are several reasons why such a system is of interest. Firstly, for some of the widely used cost functions which can be generated from (4.5), it is known that in the optimal policy it is always the case that the server begins to turn on as soon as there is a job present ($k = 1$). Secondly, due to the relatively simple nature of such a system it allows for a more elegant analysis which can be used as a stepping stone, and sanity check when moving onto more complex systems. Lastly, there may be some applications of the model where while perhaps it is optimal to allow jobs to build up in the queue, it is however not practically feasible i.e. allowing for k customers to build up at a single cash register instead of instantly having an employee begin to start serving them may be considered "bad business". Due to the exponential assumptions the queue can be represented as a CTMC and is depicted in Figure 5.2, where the state n_1, n_2 conveys that n_1 servers are on and there are n_2 jobs in the system.

5.1.1 Markov Chain Solution

Let π_{n_1,n_2} denote the steady state probability of being in state n_1, n_2. It follows from Figure 5.2 that the balance equations are:

$$(\lambda+\gamma)\pi_{0,n} = \lambda\pi_{0,n-1} \qquad (n>0) \tag{5.6}$$

$$(\lambda+\mu)\pi_{1,n} = \lambda\pi_{1,n-1} + \gamma\pi_{0,n} + \mu\pi_{1,n+1} \qquad (n>0) \tag{5.7}$$

where the boundary condition is:

$$\pi_{1,0} = \frac{\lambda}{\alpha}\pi_{0,0} \tag{5.8}$$

The first step in solving the steady state probabilities is solving the first row of the Markov chain. This is done by applying z–transforms to (5.6).

$$\begin{aligned}
& (\lambda+\gamma)\pi_{0,n} = \lambda\pi_{0,n-1} \\
\Rightarrow\quad & (\lambda+\gamma)\sum_{n=1}^{\infty}\pi_{0,n}z^n = \lambda\sum_{n=1}^{\infty}\pi_{0,n-1}z^n \\
\Rightarrow\quad & (\lambda+\gamma)\sum_{n=1}^{\infty}\pi_{0,n}z^n = \lambda z\sum_{n=0}^{\infty}\pi_{0,n}z^n \\
\Rightarrow\quad & \pi_0(z) - \pi_{0,0} = \frac{\lambda}{(\lambda+\gamma)}z\pi_0(z) \\
\Rightarrow\quad & \pi_0(z) = \frac{\pi_{0,0}}{1-\frac{\lambda}{\lambda+\gamma}z} \\
\Rightarrow\quad & \pi_0(z) = \pi_{0,0}\sum_{n=0}^{\infty}z^n\left(\frac{\lambda}{\lambda+\gamma}\right)^n
\end{aligned}$$

Taking the inverse z–transform it follows that

$$\pi_{0,n} = \pi_{0,0}\left(\frac{\lambda}{\lambda+\gamma}\right)^n. \tag{5.9}$$

The second row can be described in the form of,

$$\pi_{1,n} = Ax^n + B\left(\frac{\lambda}{\lambda+\gamma}\right)^n \tag{5.10}$$

where

$$(\lambda+\mu)x = \lambda + \mu x^2 \Rightarrow x = 1, \frac{\lambda}{\mu}.$$

B is solved by substituting Equations (5.10) and (5.9) into (5.7), which yields:

$$\begin{aligned}(\lambda+\mu)\left(Ax^n + B\left(\frac{\lambda}{\lambda+\gamma}\right)^n\right) = \lambda\left(Ax^{n-1} + B\left(\frac{\lambda}{\lambda+\gamma}\right)^{n-1}\right) + \gamma\pi_{0,0}\left(\frac{\lambda}{\lambda+\gamma}\right)^n \\ + \mu\left(Ax^{n+1} + B\left(\frac{\lambda}{\lambda+\gamma}\right)^{n+1}\right)\end{aligned} \tag{5.11}$$

$$\Rightarrow \quad (\lambda+\mu)B\frac{\lambda}{\lambda+\gamma} = \lambda B + \gamma\pi_{0,0}\frac{\lambda}{\lambda+\gamma} + \mu B\left(\frac{\lambda}{\lambda+\gamma}\right)^2$$

$$\Rightarrow \quad B\left((\lambda+\mu)\frac{\lambda}{\lambda+\gamma} - \lambda - \mu\left(\frac{\lambda}{\lambda+\gamma}\right)^2\right) = \pi_{0,0}\frac{\lambda\gamma}{\lambda+\gamma}$$

$$\Rightarrow \quad B\left(\frac{\cancel{(\lambda\gamma)}(\mu-\lambda-\gamma)}{(\lambda+\gamma)\cancel{(\lambda+\gamma)}}\right) = \pi_{0,0}\frac{\cancel{(\lambda\gamma)}}{\cancel{(\lambda+\gamma)}}$$

$$\Rightarrow \quad B = \pi_{0,0}\frac{\lambda+\gamma}{\mu-\lambda-\gamma}. \tag{5.12}$$

With B determined, all that remains is to solve for A and x. Using the definition of (5.10) and letting $n = 1$ in (5.11) gives,

$$(\lambda+\mu)Ax + (\lambda+\mu)B\left(\frac{\lambda}{\lambda+\gamma}\right) = \lambda\pi_{1,0} + \gamma\pi_{0,0}\left(\frac{\lambda}{\lambda+\gamma}\right) + \mu Ax^2 + \mu B\left(\frac{\lambda}{\lambda+\gamma}\right)^2.$$

Substituting in (5.8) and grouping terms yields,

$$A[(\lambda+\mu)x - \mu x^2] = \frac{\lambda^2}{\alpha}\pi_{0,0} + \gamma\pi_{0,0}\left(\frac{\lambda}{\lambda+\gamma}\right) + B\left(\frac{\lambda}{\lambda+\gamma}\right)\left[\frac{\mu\lambda - (\lambda+\mu)(\lambda+\gamma)}{(\lambda+\gamma)}\right].$$

From here it is seen that letting $x = 1$ or $x = \frac{\lambda}{\mu}$ gives the same result. Therefore, A can now be solved for directly by substituting for x and B.

$$\lambda A = \pi_{0,0}\left(\frac{\lambda}{\lambda+\gamma}\right)\left[\frac{\lambda(\lambda+\gamma)}{\alpha} + \gamma - \frac{\lambda^2+\lambda\gamma+\mu\gamma}{\mu-\lambda-\gamma}\right]$$

$$\Rightarrow \quad A = \pi_{0,0}\left(\frac{1}{\lambda+\gamma}\right)\left[\frac{(\lambda^2+\lambda\gamma+\gamma\alpha)(\mu-\lambda-\gamma) - \lambda^2\alpha - \lambda\gamma\alpha - \mu\gamma\alpha}{\alpha(\mu-\lambda-\gamma)}\right]$$

$$\Rightarrow \quad A = \frac{\pi_{0,0}}{\cancel{(\lambda+\gamma)}}\left[\frac{\cancel{(\lambda+\gamma)}(\lambda\mu - \lambda^2 - \lambda\gamma - \gamma\alpha - \lambda\alpha)}{\alpha(\mu-\lambda-\gamma)}\right]$$

$$\Rightarrow \quad A = \pi_{0,0}\left[\frac{\lambda}{\alpha} - \frac{\lambda+\gamma}{\mu-\lambda-\gamma}\right] \tag{5.13}$$

Substituting (5.13) and (5.12) into (5.10) gives a closed form expression for the steady state probabilities in terms of x and $\pi_{0,0}$.

$$\pi_{1,n} = \pi_{0,0}\left[\frac{\lambda}{\alpha} - \frac{\lambda+\gamma}{\mu-\lambda-\gamma}\right]x^n + \pi_{0,0}\frac{\lambda+\gamma}{\mu-\lambda-\gamma}\left(\frac{\lambda}{\lambda+\gamma}\right)^n \tag{5.14}$$

It is not clear if x is 1 or ρ. However, on inspection of (5.14) it becomes clear that letting $x = 1$ would imply $\pi_{0,0} \leq 0$, therefore it must be the case that $x = \frac{\lambda}{\mu} = \rho$.

$$\pi_{1,n} = \pi_{0,0}\left[\frac{\lambda}{\alpha}\left(\frac{\lambda}{\mu}\right)^n - \frac{\lambda+\gamma}{\mu-\lambda-\gamma}\left(\frac{\lambda}{\mu}\right)^n + \frac{\lambda+\gamma}{\mu-\lambda-\gamma}\left(\frac{\lambda}{\lambda+\gamma}\right)^n\right]$$

$$\Rightarrow \quad \pi_{1,n} = \pi_{0,0}\left[\frac{\lambda}{\alpha}\left(\frac{\lambda}{\mu}\right)^n + \frac{\lambda+\gamma}{\mu-\lambda-\gamma}\left(\left(\frac{\lambda}{\lambda+\gamma}\right)^n - \left(\frac{\lambda}{\mu}\right)^n\right)\right] \tag{5.15}$$

Now all that remains is to solve for $\pi_{0,0}$. The constraint that all steady state probabilities sum to 1 is invoked.

$$\sum_{n=0}^{\infty}\pi_{0,n} + \sum_{n=0}^{\infty}\pi_{1,n} = 1$$

$$\Rightarrow \quad \pi_{0,0}\left[\sum_{n=0}^{\infty}\left(\frac{\lambda}{\lambda+\gamma}\right)^n + \frac{\lambda}{\alpha}\sum_{n=0}^{\infty}\left(\frac{\lambda}{\mu}\right)^n + \frac{\lambda+\gamma}{\mu-\lambda-\gamma}\sum_{n=0}^{\infty}\left(\frac{\lambda}{\lambda+\gamma}\right)^n - \left(\frac{\lambda}{\mu}\right)^n\right] = 1$$

$$\Rightarrow \quad \pi_{0,0} = \left[\frac{1}{1-\frac{\lambda}{\lambda+\gamma}} + \frac{\lambda}{\alpha}\left(\frac{1}{1-\frac{\lambda}{\mu}}\right) + \frac{\lambda+\gamma}{\mu-\lambda-\gamma}\left(\frac{1}{1-\frac{\lambda}{\lambda+\gamma}} - \frac{1}{1-\frac{\lambda}{\mu}}\right)\right]^{-1}$$

$$\Rightarrow \quad \pi_{0,0} = \left[\frac{\lambda+\gamma}{\gamma} + \frac{\mu\lambda}{\alpha(\mu-\lambda)} + \cancel{\frac{\lambda+\gamma}{(\mu-\lambda-\gamma)}}\left(\frac{\lambda\cancel{(\mu-\lambda-\gamma)}}{\gamma(\mu-\lambda)}\right)\right]^{-1}$$

$$\Rightarrow \quad \pi_{0,0} = \left[\frac{\alpha(\lambda+\gamma)(\mu-\lambda) + \mu\lambda\gamma + \lambda\alpha(\lambda+\gamma)}{\gamma\alpha(\mu-\lambda)}\right]^{-1}$$

$$\Rightarrow \quad \pi_{0,0} = \frac{\alpha\gamma(\mu-\lambda)}{\mu(\alpha(\lambda+\gamma)+\lambda\gamma)}$$

$$\Rightarrow \quad \pi_{0,0} = (1-\rho)\frac{\alpha\gamma}{\alpha\gamma + \alpha\lambda + +\lambda\gamma} \tag{5.16}$$

Theorem 1. *Given an* $M/M/1 \circ \{M, M, 1\}$ *queue, described by the balance and boundary equations* (5.9), (5.15), *and* (5.16), *and* $\mu > \lambda$, *the steady-state distribution is given by:*

$$\pi_{0,n} = \pi_{0,0}\left(\frac{\lambda}{\lambda+\gamma}\right)^n$$

$$\pi_{1,n} = \pi_{0,0}\left[\frac{\lambda}{\alpha}\left(\frac{\lambda}{\mu}\right)^n + \frac{\lambda+\gamma}{\mu-\lambda-\gamma}\left(\left(\frac{\lambda}{\lambda+\gamma}\right)^n - \left(\frac{\lambda}{\mu}\right)^n\right)\right]$$

$$\pi_{0,0} = (1-\rho)\frac{\alpha\gamma}{\alpha(\lambda+\gamma)+\lambda\gamma}$$

5.1.2 Deriving System Metrics

With the CTMC solved, one can begin to work towards deriving closed form expressions for $\mathbb{E}[R]$, $\mathbb{E}[E]$, and $\mathbb{E}[Sw]$. Theorem 1 is used as a starting point to solve for $\mathbb{E}[R]$. From the distribution of the number of jobs in the system, one can arrive at $\mathbb{E}[N]$, and with Little's Law one can then arrive at $\mathbb{E}[R]$. The analysis begins by summing the steady state probabilities weighted by the number of jobs in system for

each state.

$$\mathbb{E}[N] = \sum_{n=0}^{\infty} n\pi_{0,n} + \sum_{n=0}^{\infty} n\pi_{1,n}$$

$$\Rightarrow \quad \mathbb{E}[N] = \pi_{0,0}\Bigg[\sum_{n=0}^{\infty} n\bigg(\frac{\lambda}{\lambda+\gamma}\bigg)^n + \frac{\lambda}{\alpha}\sum_{n=0}^{\infty} n\bigg(\frac{\lambda}{\mu}\bigg)^n + \frac{\lambda+\gamma}{\mu-\lambda-\gamma}\bigg(\sum_{n=0}^{\infty} n\bigg(\frac{\lambda}{\lambda+\gamma}\bigg)^n - \sum_{n=0}^{\infty} n\bigg(\frac{\lambda}{\mu}\bigg)^n\bigg)\Bigg]$$

$$\Rightarrow \quad \mathbb{E}[N] = \pi_{0,0}\Bigg[\frac{\lambda}{\lambda+\gamma}\bigg(\frac{d}{d(\frac{\lambda}{\lambda+\gamma})}\bigg(\frac{1}{1-(\frac{\lambda}{\lambda+\gamma})}\bigg)\bigg) + \frac{\lambda}{\alpha}\bigg(\frac{\lambda}{\mu}\bigg)\bigg(\frac{d}{d(\frac{\lambda}{\mu})}\bigg(\frac{1}{1-(\frac{\lambda}{\mu})}\bigg)\bigg) + \frac{\lambda+\gamma}{\mu-\lambda-\gamma}\Bigg[\frac{\lambda}{\lambda+\gamma}\bigg(\frac{d}{d(\frac{\lambda}{\lambda+\gamma})}\bigg(\frac{1}{1-(\frac{\lambda}{\lambda+\gamma})}\bigg)\bigg) + \frac{\lambda}{\mu}\bigg(\frac{d}{d(\frac{\lambda}{\mu})}\bigg(\frac{1}{1-(\frac{\lambda}{\mu})}\bigg)\bigg)\Bigg]\Bigg]$$

$$\Rightarrow \quad \mathbb{E}[N] = \pi_{0,0}\Bigg[\frac{(\frac{\lambda}{\lambda+\gamma})}{(1-(\frac{\lambda}{\lambda+\gamma}))^2} + \frac{\lambda}{\alpha}\bigg(\frac{(\frac{\lambda}{\mu})}{(1-(\frac{\lambda}{\mu}))^2}\bigg) + \frac{\lambda+\gamma}{\mu-\lambda-\gamma}\bigg(\frac{(\frac{\lambda}{\lambda+\gamma})}{(1-(\frac{\lambda}{\lambda+\gamma}))^2} - \frac{(\frac{\lambda}{\mu})}{(1-(\frac{\lambda}{\mu}))^2}\bigg)\Bigg]$$

$$\Rightarrow \quad \mathbb{E}[N] = \pi_{0,0}\Bigg[\frac{\lambda(\lambda+\gamma)}{\gamma^2} + \frac{\mu\lambda^2}{\alpha(\mu-\lambda)^2} + \frac{\lambda+\gamma}{\mu-\lambda-\gamma}\bigg(\frac{\lambda(\lambda+\gamma)(\mu-\lambda)^2 - \mu\lambda\gamma^2}{\gamma^2(\mu-\lambda)^2}\bigg)\Bigg]$$

$$\Rightarrow \quad \mathbb{E}[N] = \lambda\pi_{0,0}\Bigg[\frac{\lambda+\gamma}{\gamma^2} + \frac{\mu\lambda}{\alpha(\mu-\lambda)^2} + \frac{\lambda+\gamma}{\cancel{\mu-\lambda-\gamma}}\bigg(\frac{\cancel{(\mu-\lambda-\gamma)}(\mu\lambda-\lambda^2+\mu\gamma)}{\gamma^2(\mu-\lambda)^2}\bigg)\Bigg]$$

$$\Rightarrow \quad \mathbb{E}[N] = \lambda\pi_{0,0}\left[\frac{\alpha\lambda(\lambda+\gamma)(\mu-\lambda)^2 + \mu\lambda^2\gamma^2 + \alpha\lambda(\lambda+\gamma)(\mu\lambda+\lambda^2-\mu\gamma)}{\alpha\gamma^2(\mu-\lambda)^2}\right]$$

$$\Rightarrow \quad \mathbb{E}[N] = \mu\lambda\pi_{0,0}\left[\frac{\alpha(\lambda+\gamma)(\mu-\lambda+\gamma)+\lambda\gamma^2}{\alpha\gamma^2(\mu-\lambda)^2}\right]$$

$$\Rightarrow \quad \mathbb{E}[N] = \pi_{0,0}\frac{\mu\lambda}{\mu-\lambda}\left[\frac{(\lambda+\gamma)}{\gamma}\left(\frac{1}{\mu-\lambda}+\frac{1}{\gamma}\right)+\frac{\lambda}{\alpha(\mu-\lambda)}\right]$$

$$\Rightarrow \quad \mathbb{E}[N] = \frac{\alpha\gamma\cancel{(\mu-\lambda)}}{\mu(\alpha(\lambda+\gamma)+\lambda\gamma)}\left(\frac{\mu\lambda}{\cancel{\mu-\lambda}}\right)\left[\frac{(\lambda+\gamma)}{\gamma}\left(\frac{1}{\mu-\lambda}+\frac{1}{\gamma}\right)+\frac{\lambda}{\alpha(\mu-\lambda)}\right]$$

$$\Rightarrow \quad \mathbb{E}[N] = \frac{\alpha(\lambda+\gamma)}{\alpha(\lambda+\gamma)+\lambda\gamma}\left(\frac{\lambda}{\mu-\lambda}+\frac{\lambda}{\gamma}\right)+\frac{\lambda\gamma}{\alpha(\lambda+\gamma)+\lambda\gamma}\left(\frac{\lambda}{\mu-\lambda}\right)$$

$$\Rightarrow \quad \mathbb{E}[N] = \frac{\lambda}{\mu-\lambda}\left(\frac{\cancel{\alpha(\lambda+\gamma)+\lambda\gamma)}}{\cancel{\alpha(\lambda+\gamma)+\lambda\gamma)}}\right)+\frac{\lambda}{\gamma}\left(\frac{\alpha(\lambda+\gamma)}{\alpha(\lambda+\gamma)+\lambda\gamma}\right)$$

$$\Rightarrow \quad \mathbb{E}[N] = E[N_{M/M/1}]+\frac{\lambda}{\gamma}\left(\frac{\alpha(\lambda+\gamma)}{\alpha\gamma+\alpha\lambda+\lambda\gamma}\right) \tag{5.17}$$

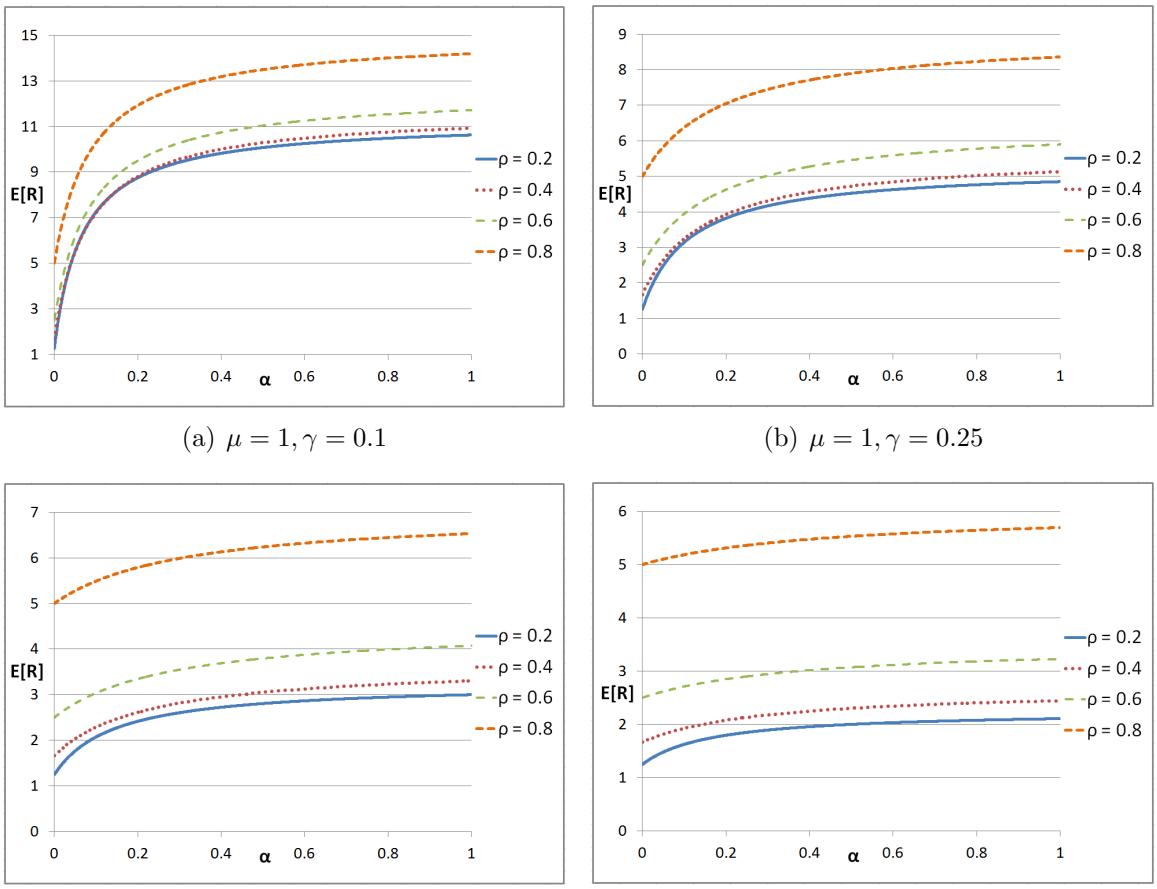
The expected response time is solved by applying Little's Law to (5.17).

$$\mathbb{E}[R] = \mathbb{E}[R_{M/M/1}]+\frac{1}{\gamma}\left(\frac{\alpha(\lambda+\gamma)}{\alpha\gamma+\alpha\lambda+\lambda\gamma}\right) \tag{5.18}$$

It is seen that $\mathbb{E}[N]$ and $\mathbb{E}[R]$ both are decompositions, where one term is the classical queueing component which has no energy concerns (here an $M/M/1$ queue) and some other term weighted by α which captures the influence of the energy-aware portion

of the queue (here it is the idling, and setup time distributions, $\{M, M, 1\}$). This makes sense as the interpretation of $\alpha = 0$ is that the server never turns off, in which case it would behave identically to an $M/M/1$ queue. On the other hand, as $\alpha \to \infty$, which means the server immediately switches off when it idles, the values of $\mathbb{E}[N]$ and $\mathbb{E}[R]$ are equal to that of an $M/M/1$ queue summed with the terms of $\frac{\lambda}{\gamma}$ and $\frac{1}{\gamma}$ respectively as seen in (5.17) and (5.18). It is also noted that $\mathbb{E}[N]$ and $\mathbb{E}[R]$ both increase in α. Taking these observations of the relationship of these metrics to the feasible range of α, one can conclude that when choosing an α and holding other parameters constant, $\mathbb{E}[N]$ and $\mathbb{E}[R]$ have lower and upper bounds and increasing α scales the metrics increasingly between these two bounds.

The idling time is chosen as the decision variable in this setting because managers usually do not have control over parameters such as the server turn on time or the arrival rate. On the other hand, they typically have control of when or how often to turn a server off. Due to this, it is important to understand how the choice of α impacts the system under varying conditions. Figure 5.3 shows $\mathbb{E}[R]$ versus α for several systems under different configurations. Looking at each sub-figure (a)-(d) individually, the shape of each curve is relatively the same across all system loads, and is shifted up the y-axis as ρ increases. These curves are similar because the difference in the upper and lower bounds of $\mathbb{E}[R]$ is the constant value $\frac{1}{\gamma}$. The difference is that the higher the system load (the greater λ is), the slower the system will approach its upper bound as α increases. However, this difference is not large as can be observed across all sub-figures of Figure 5.3. Comparing sub-figures allows one to see



(a) $\mu = 1, \gamma = 0.1$ (b) $\mu = 1, \gamma = 0.25$

(c) $\mu = 1, \gamma = 0.5$ (d) $\mu = 1, \gamma = 1$

Figure 5.3: $M/M/1 \circ \{M, M, 1\}$ response time vs α for varying γ values

the impact that the turn on times have on $\mathbb{E}[R]$. As one would expect, the lower the setup rate, the larger the range between the two bounds. However, something perhaps more surprising is how fast $\mathbb{E}[R]$ increases with α. Looking at Figure 5.3-(a), where the distance between the bounds is $1/0.1 = 10$, moving from a configuration where the server always remains on, to a configuration where the idling time rate is only 0.1, has a significant impact on $\mathbb{E}[R]$. In fact $\mathbb{E}[R]$ increases by at least 5 (50% of the distance between the bounds) over all system loads. From this, one can make the interesting conclusion that if the setup time is relatively large, having the server

have even a small chance to turn off when it idles can have a drastic (and perhaps unfavourable) effect on the expected response time of a job.

With $\mathbb{E}[N]$ and $\mathbb{E}[R]$ solved for, the analysis moves on to $\mathbb{E}[E]$. This is achieved by summing the steady state probabilities of being in the energy states *SETUP*, *BUSY* and *IDLE* defined in Section 4.1, weighted by their corresponding energy values. These steady state probabilities are denoted by, π_{Setup}, π_{Busy}, and π_{Idle} respectively.

$$\mathbb{E}[E] = E_{Busy}\pi_{Busy} + E_{Idle}\pi_{Idle} + E_{Setup}\pi_{Setup} \tag{5.19}$$

One can start by making the simple observation that for any stable single server system,

$$\pi_{Busy} = \rho. \tag{5.20}$$

Also, there is only one system state in the energy state *IDLE*, so it follows that,

$$\pi_{Idle} = \pi_{1,0} = (1-\rho)\frac{\lambda\gamma}{\alpha\gamma + \alpha\lambda + \lambda\gamma}. \tag{5.21}$$

So to solve for $\mathbb{E}[E]$, all that remains is to solve for π_{Setup}. This is done by summing all steady state probabilities in the first row of the Markov chain, excluding $\pi_{0,0}$.

$$\begin{aligned} & \pi_{Setup} = \sum_{n=1}^{\infty} \pi_{0,n} \\ \Rightarrow \quad & \pi_{Setup} = \sum_{n=0}^{\infty} \pi_{0,n} - \pi_{0,0} \\ \Rightarrow \quad & \pi_{Setup} = \pi_{0,0}\left(\sum_{n=0}^{\infty}\left(\frac{\lambda}{\lambda+\gamma}\right)^n - 1\right) \end{aligned}$$

$$\begin{aligned}
\Rightarrow \quad & \pi_{Setup} = \pi_{0,0}\left(\frac{1}{1-\frac{\lambda}{\lambda+\gamma}} - 1\right) \\
\Rightarrow \quad & \pi_{Setup} = \frac{\lambda}{\gamma}\pi_{0,0} \\
\Rightarrow \quad & \pi_{Setup} = (1-\rho)\frac{\lambda\alpha}{\alpha(\lambda+\gamma)+\lambda\gamma} \qquad (5.22)
\end{aligned}$$

Substituting 5.20, 5.21, and 5.22 into (5.19) gives a closed form expression for the expected energy.

$$\mathbb{E}[E] = \rho E_{Busy} + E_{Idle}(1-\rho)\frac{\lambda\gamma}{\alpha\gamma+\alpha\lambda+\lambda\gamma} + E_{Setup}(1-\rho)\frac{\alpha\lambda}{\alpha(\lambda+\gamma)+\lambda\gamma}$$

$$\Rightarrow \quad \mathbb{E}[E] = E_{Busy}\left[\rho + r_{Idle}(1-\rho)\frac{\lambda\gamma}{\alpha(\lambda+\gamma)+\lambda\gamma} + r_{Setup}(1-\rho)\frac{\alpha\lambda}{\alpha(\lambda+\gamma)+\lambda\gamma}\right]$$

$$\Rightarrow \quad \mathbb{E}[E] = E_{Busy}\left[\rho + (1-\rho)\frac{\lambda}{\alpha(\lambda+\gamma)+\lambda\gamma}(r_{Idle}\gamma + r_{Setup}\alpha)\right]$$

$$\begin{aligned}
\Rightarrow \quad \mathbb{E}[E] = E_{Busy}\Bigg[\rho + (1-\rho)\frac{\alpha\gamma+\alpha\lambda+\lambda\gamma}{\alpha\gamma+\alpha\lambda+\lambda\gamma}r_{Idle} \\
+ (1-\rho)\frac{\alpha}{\alpha\gamma+\alpha\lambda+\lambda\gamma}(\lambda r_{Setup} - (\lambda+\gamma)r_{Idle})\Bigg]
\end{aligned}$$

$$\Rightarrow \quad \mathbb{E}[E] = \mathbb{E}[E_{M/M/1}] + E_{Busy}(1-\rho)\frac{\alpha}{\alpha\gamma+\alpha\lambda+\lambda\gamma}(\lambda r_{Setup} - (\lambda+\gamma)r_{Idle})$$

This gives us the true expected energy usage of the system, however since in the cost functions (4.5), $\mathbb{E}[E]$ is weighted by a constant β, the constant E_{Busy} can be absorbed,

and a metric normalized by this weight can be derived.

$$\mathbb{E}[E^N] = \mathbb{E}[E^N_{M/M/1}] + (1-\rho)\frac{\alpha}{\alpha\gamma + \alpha\lambda + \lambda\gamma}(\lambda r_{Setup} - (\lambda+\gamma)r_{Idle}). \tag{5.23}$$

The final metric is the simplest to solve for. The expected rate of switching is equal to the product of α and the steady state probability of being idle, $\pi_{1,0}$.

$$\begin{aligned} & \mathbb{E}[Sw] = \alpha\pi_{1,0} \\ \Rightarrow \quad & \mathbb{E}[Sw] = (1-\rho)\frac{\alpha\lambda\gamma}{\alpha\gamma + \alpha\lambda + \lambda\gamma} \end{aligned} \tag{5.24}$$

While there is more analysis to be done on equations (5.23) and (5.24), this is deferred to Section 5.3 as the reader will see conclusions about these expressions can be made in a more general setting.

With (5.18), (5.23), and (5.24), system metrics for the $M/M/1 \circ \{M, M, 1\}$ are fully determined. However, as stated previously in this section, although one can derive optimal policies within the set of policies described by the $M/M/1 \circ \{M, M, 1\}$ queue, one cannot derive the optimal policy for a single server system (even under full exponential assumptions), under most non-trivial cost functions. This is due to the fact that the $M/M/1 \circ \{M, M, 1\}$ queue does not account for jobs accumulating before entering *SETUP*. In order to arrive at an optimal policy under any cost function, a more general system must be analysed, which is done in the following section.

5.2 The M/M/1 ∘ {M,M,k} Queue

Here the exact analysis of the $M/M/1 \circ \{M, M, k\}$ queue is presented. As was seen in the previous section, the analysis of the $M/M/1 \circ \{M, M, 1\}$ queue fell short when determining the optimal policy. Although in this section all assumptions on the underlying distributions are unchanged, the $M/M/1 \circ \{M, M, k\}$ allows for jobs to accumulate in the queue before entering the state *SETUP*. For the reasons discussed in Section 4.3.1, this small change to the system now allows for the model to describe the optimal policy under any cost function of the form (4.5).

5.2.1 Markov Chain Solution

Again, due to the assumption that all underlying random variables are exponentially distributed, the system can be modelled as a CTMC, as seen in Figure 5.4. The same notation for the states of the form (n_1, n_2), is kept from Section 5.1, where n_1 is 0 if the server is off, and 1 if it is on, and n_2 denotes the number of jobs in the system. However, this Markov chain is considerably more complex than the one previously solved, as upon inspection, the two rows do not begin to repeat until the after the kth column. This leads to having four clear sections of the CTMC, which yields the following balance equations:

$$\pi_{0,n} = \pi_{0,0} \qquad (n < k) \tag{5.25}$$

$$(\lambda + \gamma)\pi_{0,n} = \lambda\pi_{0,n-1} \qquad (n \geq k) \tag{5.26}$$

$$\mu\pi_{1,n} = \lambda\pi_{1,n-1} + \lambda\pi_{0,n-1} \qquad (0 < n < k) \tag{5.27}$$

$$(\mu + \lambda)\pi_{1,n} = \lambda\pi_{0,n-1} + \gamma\pi_{0,n} + \mu\pi_{1,n+1} \qquad (n \geq k) \tag{5.28}$$

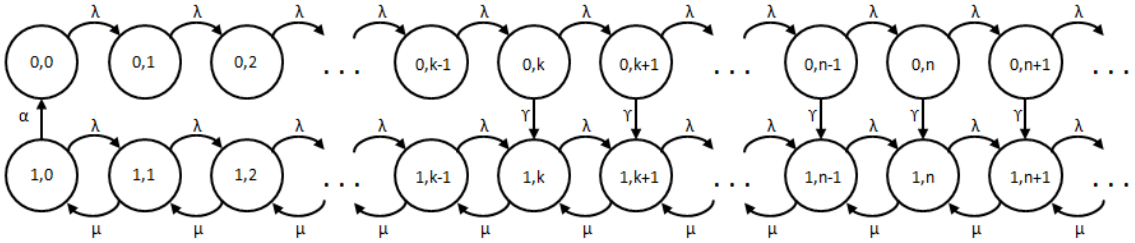


Figure 5.4: $M/M/1 \circ \{M, M, k\}$ Queue

and the boundary equation is:

$$\pi_{1,0} = \frac{\lambda}{\alpha}\pi_{0,0}. \tag{5.29}$$

The steady state probabilities of the first section of the Markov chain, where $n_1 = 0$ and $0 \leq n_2 < k$, are trivially solved with respect to $\pi_{0,0}$ using (5.25). This follows so simply because the only action performed in these states is a job arriving to the system.

The recursion observed for the first row in the Markov chain shown in Figure 5.2 is seen again for the steady state probabilities of the repeating section of the first row, described by (5.26). The difference here is the repeating portion of the first row does not occur until the state $(0, k)$, so the base case of the recursion is $\pi_{0,k-1}$. However, due to the first balance equation, it is known that $\pi_{0,k-1} = \pi_{0,0}$, and the steady state probabilities of the repeating section of the first row are thus given by:

$$\pi_{0,n} = \pi_{0,0}\left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)} \qquad (k \leq n). \tag{5.30}$$

With the steady state probabilities of the first row now solved for (with respect to

$\pi_{0,0}$), the analysis proceeds to solve for the steady state probabilities of the second row, which is a more challenging problem, as will be seen. To solve for the first section of the second row, where $n_1 = 1$ and $0 \leq n_2 < k$, the balance equation (5.27) was carefully chosen. Instead of looking at the rate in and rate out of every state (which leads to an overflow into the unknown probabilities of the repeating component), one can choose to look at the rate in and rate out between the columns of the CTMC. This allows the exploitation of the simplicity of the steady state probabilities in the non-repeating portion of the first row. Furthermore, using this balance equation, all the steady state probabilities in question can be described without any of the steady state values of the repeating portion. Using (5.27) to solve each of these probabilities, one can build up to a finite sum, and arrive at a general closed form solution after applying the boundary condition of (5.29).

$$
\begin{aligned}
& \mu\pi_{1,1} = \lambda\pi_{1,0} + \lambda\pi_{0,0} \\
\Rightarrow \quad & \pi_{1,1} = \rho\pi_{1,0} + \rho\pi_{0,0}
\end{aligned}
$$

$$
\begin{aligned}
& \mu\pi_{1,2} = \lambda\pi_{1,1} + \lambda\pi_{0,0} \\
\Rightarrow \quad & \pi_{1,2} = \rho^2\pi_{1,0} + \rho^2\pi_{0,0} + \rho\pi_{0,0} \\
& \vdots \\
& \mu\pi_{1,n} = \lambda\pi_{1,n-1} + \lambda\pi_{0,0} && (0 < n < k) \\
\Rightarrow \quad & \pi_{1,n} = \rho^n\pi_{1,0} + \pi_{0,0}\sum_{i=1}^{n}\rho^i && (0 < n < k) \\
\Rightarrow \quad & \pi_{1,n} = \pi_{0,0}\left(\frac{\lambda}{\alpha}\rho^n + \rho\frac{1-\rho^n}{1-\rho}\right) && (0 < n < k) \qquad (5.31)
\end{aligned}
$$

Now all that remains to solve the steady state probabilities with respect to $\pi_{0,0}$ is to solve the repeating portion of the second row. Similar to the analysis in Section 5.1, this part of the Markov chain can be described in the form,

$$\pi_{1,n} = Ax^{n-(k-1)} + B\left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)}, \tag{5.32}$$

where again, $x = 1, \rho$. Substituting (5.32) and (5.30) into (5.28) yields,

$$\begin{aligned}(\lambda+\mu)\left(Ax^{n-(k-1)} + B\left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)}\right) = \lambda\left(Ax^{n-k} + B\left(\frac{\lambda}{\lambda+\gamma}\right)^{n-k}\right) \\ + \gamma\pi_{0,0}\left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)} \\ + \mu\left(Ax^{n+2-k} + B\left(\frac{\lambda}{\lambda+\gamma}\right)^{n+2-k}\right)\end{aligned} \tag{5.33}$$

and from which, one can separate and solve for B.

$$(\lambda+\mu)B\frac{\lambda}{\lambda+\gamma} = \lambda B + \gamma\pi_{0,0}\frac{\lambda}{\lambda+\gamma} + \mu B\left(\frac{\lambda}{\lambda+\gamma}\right)^2$$

$$\Rightarrow \quad B\left((\lambda+\mu)\frac{\lambda}{\lambda+\gamma} - \lambda - \mu\left(\frac{\lambda}{\lambda+\gamma}\right)^2\right) = \pi_{0,0}\frac{\lambda\gamma}{\lambda+\gamma}$$

$$\Rightarrow \quad B\left(\frac{\cancel{(\lambda\gamma)}(\mu-\lambda-\gamma)}{(\lambda+\gamma)\cancel{(\lambda+\gamma)}}\right) = \pi_{0,0}\frac{\cancel{(\lambda\gamma)}}{\cancel{(\lambda+\gamma)}}$$

$$\Rightarrow \quad B = \pi_{0,0}\frac{\lambda+\gamma}{\mu-\lambda-\gamma} \tag{5.34}$$

It is noted that (5.34) is equivalent to (5.12). In other words, the value for B (the non-homogeneous coefficient) in the $M/M/1 \circ \{M, M, 1\}$ analysis equals the value of B in the $M/M/1 \circ \{M, M, k\}$ analysis. This is perhaps not surprising, as B is the coefficient of the non-homogeneous component (values from the first row of the Markov chain) in (5.32) and the recursion on the repeating portion of the first row of the Markov chain of an $M/M/1 \circ \{M, M, k\}$ is the same as that of an $M/M/1 \circ \{M, M, 1\}$ queue.

With B solved, the analysis continues by solving for A. This is done by letting $n = k$ in (5.33) and using the definition (5.32) to reveal a term involving $\pi_{1,k-1}$.

$$\lambda A = \lambda \pi_{1,k-1} + \gamma \pi_{0,0} \frac{\lambda}{\lambda + \gamma} + B \frac{\lambda}{\lambda + \gamma} \left(\frac{\mu\lambda - (\mu + \lambda)(\lambda + \gamma)}{\lambda + \gamma} \right)$$

Substituting in (5.31) for $n = k - 1$ gives

$$A = \frac{\pi_{0,0}}{\lambda + \gamma} \left[\frac{\lambda(\lambda + \gamma)}{\alpha} \rho^{k-1} + \rho(\lambda + \gamma) \frac{1 - \rho^{k-1}}{1 - \rho} + \gamma - \frac{\lambda^2 + \lambda\gamma + \mu\gamma}{\mu - \lambda - \gamma} \right]$$

$$\Rightarrow \quad A = \frac{\pi_{0,0}}{\lambda + \gamma} \left[\frac{\lambda(\lambda + \gamma)}{\alpha} \rho^{k-1} + \rho(\lambda + \gamma) \frac{1 - \rho^{k-1}}{1 - \rho} + \frac{\cancel{\gamma\mu} - \cancel{\mu\gamma} - (\lambda + \gamma)^2}{\mu - \lambda - \gamma} \right]$$

$$\Rightarrow \quad A = \pi_{0,0} \left[\frac{\lambda}{\alpha} \rho^{k-1} + \rho \frac{1 - \rho^{k-1}}{1 - \rho} - \frac{\lambda + \gamma}{\mu - \lambda - \gamma} \right]$$

$$\Rightarrow \quad A = \pi_{0,0} \left[\left(\frac{\lambda}{\alpha} - \frac{\lambda}{\mu - \lambda} \right) \rho^{k-1} + \frac{\lambda}{\mu - \lambda} - \frac{\lambda + \gamma}{\mu - \lambda - \gamma} \right]$$

$$\Rightarrow \quad A = \pi_{0,0}\left[\left(\frac{\lambda}{\alpha} - \frac{\lambda}{\mu-\lambda}\right)\rho^{k-1} - \frac{\mu\gamma}{(\mu-\lambda)(\mu-\lambda-\gamma)}\right] \tag{5.35}$$

From here one can solve for the steady state probabilities of the repeating portion of the second row by substituting (5.35), (5.34), and $x = \rho$ into (5.32) and simplifying.

$$\pi_{1,n} = \pi_{0,0}\left[\left(\frac{\lambda}{\alpha} - \frac{\lambda}{\mu-\lambda}\right)\rho^{n} - \frac{\mu\gamma}{(\mu-\lambda)(\mu-\lambda-\gamma)}\rho^{n-(k-1)} + \frac{\lambda+\gamma}{\mu-\lambda-\gamma}\left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)}\right]$$

$$\Rightarrow \quad \pi_{1,n} = \pi_{0,0}\left[\left(\frac{\lambda}{\alpha} - \frac{\lambda}{\mu-\lambda}\right)\rho^{n} + \frac{1}{\mu-\lambda-\gamma}\left((\lambda+\gamma)\left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)} - \frac{\gamma}{1-\rho}\rho^{n-(k-1)}\right)\right] \tag{5.36}$$

Now all that remains to completely determine the steady state distribution is to solve for $\pi_{0,0}$. The typical approach of exploiting the fact that all probabilities sum to 1 is used.

$$1 = \sum_{n=0}^{k-1}\pi_{0,n} + \sum_{n=k}^{\infty}\pi_{0,n} + \sum_{n=0}^{k-1}\pi_{1,n} + \sum_{n=k}^{\infty}\pi_{1,n}$$

$$\Rightarrow \quad 1 = \pi_{0,0}\sum_{n=0}^{k-1}1 + \pi_{0,0}\sum_{n=k}^{\infty}\left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)} + \pi_{0,0}\sum_{n=0}^{k-1}\left(\frac{\lambda}{\alpha}\rho^{n} + \frac{\lambda}{\mu-\lambda}(1-\rho^{n})\right) + \pi_{0,0}\sum_{n=k}^{\infty}\left[\left(\frac{\lambda}{\alpha} - \frac{\lambda}{\mu-\lambda}\right)\rho^{n} + \frac{1}{\mu-\lambda-\gamma}\left((\lambda+\gamma)\left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)} - \frac{\gamma}{1-\rho}\rho^{n-(k-1)}\right)\right]$$

$$\Rightarrow \quad \pi_{0,0} = \left[k + \sum_{n=0}^{\infty} \left(\frac{\lambda}{\lambda+\gamma} \right)^n - 1 + \left(\frac{\lambda}{\alpha} - \frac{\lambda}{\mu-\lambda} \right) \sum_{n=0}^{\infty} \rho^n + \frac{\lambda}{\mu-\lambda} k \right.$$
$$+ \frac{\lambda+\gamma}{\mu-\lambda-\gamma} \left(\sum_{n=0}^{\infty} \left(\frac{\lambda}{\lambda+\gamma} \right)^n - 1 \right)$$
$$\left. - \frac{\mu\gamma}{(\mu-\lambda)(\mu-\lambda-\gamma)} \left(\sum_{n=0}^{\infty} \rho^n - 1 \right) \right]^{-1}$$

$$\Rightarrow \quad \pi_{0,0} = \left[\frac{\mu k}{\mu-\lambda} + \frac{1}{1-\frac{\lambda}{\lambda+\gamma}} - 1 + \frac{\lambda(\mu-\lambda-\alpha)}{\alpha(\mu-\lambda)} \left(\frac{1}{1-\frac{\lambda}{\mu}} \right) \right.$$
$$\left. + \frac{\lambda+\gamma}{\mu-\lambda-\gamma} \left(\frac{1}{1-\frac{\lambda}{\lambda+\gamma}} - 1 \right) - \frac{\mu\gamma}{(\mu-\lambda)(\mu-\lambda-\gamma)} \left(\frac{1}{1-\frac{\lambda}{\mu}} - 1 \right) \right]^{-1}$$

$$\Rightarrow \quad \pi_{0,0} = \left[\frac{\mu k}{\mu-\lambda} + \frac{\lambda}{\gamma} + \frac{\mu\lambda(\mu-\lambda-\alpha)}{\alpha(\mu-\lambda)^2} + \frac{\lambda(\lambda+\gamma)}{\gamma(\mu-\lambda-\gamma)} - \frac{\mu\lambda\gamma}{(\mu-\lambda)^2(\mu-\lambda-\gamma)} \right]^{-1}$$

$$\Rightarrow \quad \pi_{0,0} = \frac{\mu-\lambda}{\mu} \left[k + \frac{\lambda(\mu-\lambda)}{\mu\gamma} + \frac{\lambda(\mu-\lambda-\alpha)}{\alpha(\mu-\lambda)} + \frac{\lambda(\lambda+\gamma)(\mu-\lambda)^2 - \mu\lambda\gamma^2}{\mu\gamma(\mu-\lambda)(\mu-\lambda-\gamma)} \right]^{-1}$$

$$\Rightarrow \quad \pi_{0,0} = (1-\rho) \left[k + \frac{\lambda(\mu-\lambda)}{\mu\gamma} + \frac{\lambda(\mu-\lambda-\alpha)}{\alpha(\mu-\lambda)} + \frac{\lambda(\mu\gamma+\mu\lambda-\lambda^2)\cancel{(\mu-\lambda-\gamma)}}{\mu\gamma(\mu-\lambda)\cancel{(\mu-\lambda-\gamma)}} \right]^{-1}$$

$$\Rightarrow \quad \pi_{0,0} = (1-\rho) \left[k + \frac{\lambda(\alpha(\mu-\lambda)^2 + \mu\gamma(\mu-\lambda-\alpha) + \alpha(\mu\gamma+\mu\lambda-\lambda^2))}{\mu\alpha\gamma(\mu-\lambda)} \right]^{-1}$$

$$\Rightarrow \quad \pi_{0,0} = (1-\rho) \left[k + \frac{\mu\lambda(\alpha\mu - 2\alpha\lambda + \mu\gamma - \lambda\gamma - \cancel{\alpha\gamma} + \cancel{\alpha\gamma} + \cancel{\alpha\lambda}) + \cancel{\alpha\lambda^3} - \cancel{\alpha\lambda^3}}{\mu\alpha\gamma(\mu-\lambda)} \right]^{-1}$$

$$\Rightarrow \quad \pi_{0,0} = (1-\rho)\left[k + \frac{\lambda(\alpha+\gamma)\cancel{(\mu-\lambda)}}{\alpha\gamma\cancel{(\mu-\lambda)}}\right]^{-1}$$

$$\Rightarrow \quad \pi_{0,0} = (1-\rho)\frac{\alpha\gamma}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} \tag{5.37}$$

Theorem 2. *The steady state distribution of an* $M/M/1 \circ \{M, M, k\}$ *queue is given by,*

$$\pi_{0,n} = \pi_{0,0}, \qquad \pi_{1,n} = \pi_{0,0}\left(\frac{\lambda}{\alpha}\rho^n + \frac{\lambda}{\mu-\lambda}(1-\rho^n)\right)$$

for $(0 \leq n < k)$*,*

$$\pi_{0,n} = \pi_{0,0}\left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)}$$

and,

$$\pi_{1,n} = \pi_{0,0}\left[\left(\frac{\lambda}{\alpha} - \frac{\lambda}{\mu-\lambda}\right)\rho^n + \frac{1}{\mu-\lambda-\gamma}\left((\lambda+\gamma)\left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)} - \frac{\gamma}{1-\rho}\rho^{n-(k-1)}\right)\right]$$

for $(k \leq n)$*, where*

$$\pi_{0,0} = (1-\rho)\frac{\alpha\gamma}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}.$$

5.2.2 Deriving System Metrics

As usual, from this point the analysis proceeds to solve for $\mathbb{E}[N]$ with the goal of arriving at $\mathbb{E}[R]$. The typical approach of summing weighted steady state probabilities

is used.

$$\mathbb{E}[N] = \sum_{n=0}^{k-1} n\pi_{0,n} + \sum_{n=k}^{\infty} n\pi_{0,n} + \sum_{n=0}^{k-1} n\pi_{1,n} + \sum_{n=k}^{\infty} n\pi_{1,n}$$

$$\begin{aligned}
\Rightarrow \quad \mathbb{E}[N] &= \pi_{0,0} \sum_{n=0}^{k-1} n + \pi_{0,0} \sum_{n=k}^{\infty} n \left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)} \\
&\quad + \pi_{0,0} \sum_{n=0}^{k-1} n \left(\frac{\lambda}{\alpha}\rho^n + \frac{\lambda}{\mu-\lambda}(1-\rho^n)\right) + \pi_{0,0} \sum_{n=k}^{\infty} n \left[\left(\frac{\lambda}{\alpha} - \frac{\lambda}{\mu-\lambda}\right)\rho^n \right. \\
&\quad \left. + \frac{1}{\mu-\lambda-\gamma}\left((\lambda+\gamma)\left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)} - \frac{\gamma}{1-\rho}\rho^{n-(k-1)}\right)\right]
\end{aligned}$$

$$\begin{aligned}
\Rightarrow \quad \frac{\mathbb{E}[N]}{\pi_{0,0}} &= \sum_{n=0}^{k-1} n + \sum_{n=k}^{\infty} n \left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)} + \frac{\lambda}{\alpha} \sum_{n=0}^{\infty} n\rho^n + \frac{\lambda}{\mu-\lambda} \sum_{n=0}^{k-1} n \\
&\quad - \frac{\lambda}{\mu-\lambda} \sum_{n=0}^{\infty} n\rho^n + \frac{\lambda+\gamma}{\mu-\lambda-\gamma} \sum_{n=k}^{\infty} n \left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)} \\
&\quad - \frac{\mu\gamma}{(\mu-\lambda)(\mu-\lambda-\gamma)} \sum_{n=k}^{\infty} n\rho^{n-(k-1)}
\end{aligned} \tag{5.38}$$

Due to the complexity of these expressions, some of the terms are solved separately to keep the algebra clean. These terms are denoted by T_1, T_2, and T_3.

$$\begin{aligned}
& T_1 \triangleq \sum_{n=k}^{\infty} n \left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)} \\
\Rightarrow \quad & T_1 = \sum_{n=1}^{\infty} (n + (k-1)) \left(\frac{\lambda}{\lambda+\gamma}\right)^{n} \\
\Rightarrow \quad & T_1 = \sum_{n=1}^{\infty} n \left(\frac{\lambda}{\lambda+\gamma}\right)^{n} + (k-1) \sum_{n=1}^{\infty} \left(\frac{\lambda}{\lambda+\gamma}\right)^{n}
\end{aligned}$$

$$\Rightarrow \quad T_1 = \frac{\lambda}{\lambda+\gamma}\sum_{n=0}^{\infty} n\left(\frac{\lambda}{\lambda+\gamma}\right)^{n-1} + (k-1)\left[\sum_{n=0}^{\infty}\left(\frac{\lambda}{\lambda+\gamma}\right)^{n} - 1\right]$$

$$\Rightarrow \quad T_1 = \frac{\lambda}{\lambda+\gamma}\frac{d}{d(\frac{\lambda}{\lambda+\gamma})}\sum_{n=0}^{\infty}\left(\frac{\lambda}{\lambda+\gamma}\right)^{n} + \frac{k-1}{1-\frac{\lambda}{\lambda+\gamma}} - (k-1)$$

$$\Rightarrow \quad T_1 = \frac{\lambda}{\lambda+\gamma}\frac{1}{(1-\frac{\lambda}{\lambda+\gamma})^2} + \frac{(\lambda+\gamma)(k-1)-\gamma(k-1)}{\gamma}$$

$$\Rightarrow \quad T_1 = \frac{\lambda(\lambda+\gamma)}{\gamma^2} + \frac{\lambda(k-1)}{\gamma}$$

$$\Rightarrow \quad T_1 = \frac{\lambda(\lambda+k\gamma)}{\gamma^2} \tag{5.39}$$

$$T_2 \triangleq \sum_{n=k}^{\infty} n\rho^{n-(k-1)}$$

$$\Rightarrow \quad T_2 = \sum_{n=1}^{\infty}(n+(k-1))\rho^n$$

$$\Rightarrow \quad T_2 = \sum_{n=1}^{\infty} n\rho^n + (k-1)\sum_{n=1}^{\infty}\rho^n$$

$$\Rightarrow \quad T_2 = \rho\sum_{n=1}^{\infty} n\rho^{n-1} + \frac{k-1}{1-\rho} - (k-1)$$

$$\Rightarrow \quad T_2 = \rho\frac{d}{d\rho}\frac{1}{1-\rho} + \frac{\mu(k-1)-(\mu-\lambda)(k-1)}{\mu-\lambda}$$

$$\Rightarrow \quad T_2 = \frac{\lambda}{\mu}\frac{1}{(1-\frac{\lambda}{\mu})^2} + \frac{\lambda(k-1)}{\mu-\lambda}$$

$$\Rightarrow \quad T_2 = \frac{\mu\lambda}{(\mu-\lambda)^2} + \frac{\lambda(k-1)}{\mu-\lambda}$$

$$\Rightarrow \quad T_2 = \frac{\lambda(k(\mu-\lambda)+\lambda)}{(\mu-\lambda)^2} \tag{5.40}$$

$$T_3 \triangleq \sum_{n=0}^{\infty} n\rho^n$$

$$
\begin{aligned}
\Rightarrow \quad & T_3 = \rho \sum_{n=0}^{\infty} n\rho^{n-1} \\
\Rightarrow \quad & T_3 = \rho \frac{d}{d\rho} \sum_{n=0}^{\infty} \rho^n \\
\Rightarrow \quad & T_3 = \rho \frac{d}{d\rho} \frac{1}{1-\rho} \\
\Rightarrow \quad & T_3 = \frac{\mu\lambda}{(\mu-\lambda)^2} \qquad (5.41)
\end{aligned}
$$

Substituting (5.39), (5.40), and (5.41) into (5.38) yields,

$$
\begin{aligned}
\frac{\mathbb{E}[N]}{\pi_{0,0}} = & \frac{k(k-1)}{2} + \frac{\lambda(\lambda+k\gamma)}{\gamma^2} + \frac{\lambda}{\alpha}\left(\frac{\mu\lambda}{(\mu-\lambda)^2}\right) + \frac{\lambda}{\mu-\lambda}\left(\frac{k(k-1)}{2}\right) \\
& - \frac{\lambda}{\mu-\lambda}\left(\frac{\mu\lambda}{(\mu-\lambda)^2}\right) + \frac{\lambda+\gamma}{\mu-\lambda-\gamma}\left(\frac{\lambda(\lambda+k\gamma)}{\gamma^2}\right) \\
& - \frac{\mu\gamma}{(\mu-\lambda)(\mu-\lambda-\gamma)}\left(\frac{\lambda(k(\mu-\lambda)+\lambda)}{(\mu-\lambda)^2}\right)
\end{aligned}
$$

$$
\begin{aligned}
\Rightarrow \quad \frac{\mathbb{E}[N]}{\pi_{0,0}} = & \frac{k(k-1)}{2}\left(1 + \frac{\lambda}{\mu-\lambda}\right) + \frac{\lambda(\lambda+k\gamma)}{\gamma^2}\left(1 + \frac{\lambda+\gamma}{\mu-\lambda-\gamma}\right) \\
& + \frac{\mu\lambda}{(\mu-\lambda)^2}\left(\frac{\lambda}{\alpha} - \frac{\lambda}{\mu-\lambda}\right) - \frac{\mu\lambda\gamma(k(\mu-\lambda)+\lambda)}{(\mu-\lambda)^3(\mu-\lambda-\gamma)}
\end{aligned}
$$

$$
\Rightarrow \quad \frac{\mathbb{E}[N]}{\pi_{0,0}} = \frac{\mu k(k-1)}{2(\mu-\lambda)} + \frac{\mu\lambda(\lambda+k\gamma)}{\gamma^2(\mu-\lambda-\gamma)} + \frac{\mu\lambda^2(\mu-\lambda-\alpha)}{\alpha(\mu-\lambda)^3} - \frac{\mu\lambda\gamma(k(\mu-\lambda)+\lambda)}{(\mu-\lambda)^3(\mu-\lambda-\gamma)}
$$

$$
\begin{aligned}
\Rightarrow \quad \frac{\mathbb{E}[N]}{\pi_{0,0}} = & \frac{\mu k(k-1)}{2(\mu-\lambda)} + \frac{\mu\lambda(\lambda+k\gamma)}{\gamma^2(\mu-\lambda-\gamma)} \\
& + \frac{\mu\lambda^2(\mu-\lambda-\alpha)(\mu-\lambda-\gamma) - \mu\alpha\lambda\gamma(k(\mu-\lambda)+\lambda)}{\alpha(\mu-\lambda)^3(\mu-\lambda-\gamma)}
\end{aligned}
$$

$$\Rightarrow \quad \frac{\mathbb{E}[N]}{\pi_{0,0}} = \frac{\mu k(k-1)}{2(\mu-\lambda)} + \frac{\mu\lambda(\lambda+k\gamma)}{\gamma^2(\mu-\lambda-\gamma)} + \mu\lambda\frac{\mu^2\lambda - 2\mu\lambda^2 - \mu\lambda\gamma + \lambda^3 + \lambda^2\gamma}{\alpha(\mu-\lambda)^3(\mu-\lambda-\gamma)} - \mu\lambda\frac{\mu\alpha\lambda + \alpha\lambda^2 - \alpha\gamma k(\mu-\lambda) + \cancel{\alpha\lambda\gamma} - \cancel{\alpha\lambda\gamma}}{\alpha(\mu-\lambda)^3(\mu-\lambda-\gamma)}$$

$$\Rightarrow \quad \frac{\mathbb{E}[N]}{\pi_{0,0}} = \frac{\mu k(k-1)}{2(\mu-\lambda)} + \frac{\mu\lambda(\lambda+k\gamma)}{\gamma^2(\mu-\lambda-\gamma)} + \frac{\mu\lambda\cancel{(\mu-\lambda)}(\mu\lambda - \lambda^2 - \alpha\lambda - \lambda\gamma - \alpha\gamma k)}{\alpha\cancel{(\mu-\lambda)}(\mu-\lambda)^2(\mu-\lambda-\gamma)}$$

$$\Rightarrow \quad \frac{\mathbb{E}[N]}{\pi_{0,0}} = \frac{\mu k(k-1)}{2(\mu-\lambda)} + \frac{\mu\lambda[\alpha(\mu-\lambda)^2(\lambda-k\gamma) + \gamma^2(\mu\lambda - \lambda^2 - \alpha\lambda - \lambda\gamma - \alpha\gamma k)]}{\alpha\gamma^2(\mu-\lambda)^2(\mu-\lambda-\gamma)}$$

$$\Rightarrow \quad \frac{\mathbb{E}[N]}{\pi_{0,0}} = \frac{\mu k(k-1)}{2(\mu-\lambda)} + \mu\lambda\frac{\mu^2\alpha\lambda - 2\mu\alpha\lambda^2 + \alpha\lambda^3}{\alpha\gamma^2(\mu-\lambda)^2(\mu-\lambda-\gamma)} + \mu\lambda\frac{\gamma(k\mu^2\alpha - 2k\mu\alpha\lambda + k\alpha\lambda^2 + \mu\lambda\gamma - \lambda^2\gamma - \alpha\lambda\gamma - \lambda\gamma^2 - k\alpha\gamma^2)}{\alpha\gamma^2(\mu-\lambda)^2(\mu-\lambda-\gamma)}$$

$$\Rightarrow \quad \frac{\mathbb{E}[N]}{\pi_{0,0}} = \frac{\mu k(k-1)}{2(\mu-\lambda)} + \mu\lambda\frac{\mu^2\alpha\lambda - 2\mu\alpha\lambda^2 + \alpha\lambda^3 + \gamma(\mu-\lambda-\alpha)(k\alpha\gamma + \alpha\lambda + \lambda\gamma)}{\alpha\gamma^2(\mu-\lambda)^2(\mu-\lambda-\gamma)} + \mu\lambda\frac{\gamma(k\mu^2\alpha - 2k\mu\alpha\lambda + k\alpha\lambda^2 - k\mu\alpha\gamma + k\alpha\lambda\gamma - \mu\alpha\lambda + \alpha\lambda^2)}{\alpha\gamma^2(\mu-\lambda)^2(\mu-\lambda-\gamma)}$$

$$\Rightarrow \quad \frac{\mathbb{E}[N]}{\pi_{0,0}} = \frac{\mu k(k-1)}{2(\mu-\lambda)} + \frac{\mu\lambda\cancel{\gamma(\mu-\lambda-\gamma)}(k\alpha\gamma + \alpha\lambda + \lambda\gamma)}{\alpha\cancel{\gamma^2}(\mu-\lambda)^2\cancel{(\mu-\lambda-\gamma)}} + \mu\lambda\frac{\mu^2\alpha\lambda - 2\mu\alpha\lambda^2 + \alpha\lambda^3}{\alpha\gamma^2(\mu-\lambda)^2(\mu-\lambda-\gamma)} + \mu\lambda\frac{\gamma(k\mu^2\alpha - 2k\mu\alpha\lambda + k\alpha\lambda^2 - k\mu\alpha\gamma + k\alpha\lambda\gamma - \mu\alpha\lambda + \alpha\lambda^2)}{\alpha\gamma^2(\mu-\lambda)^2(\mu-\lambda-\gamma)}$$

$$\Rightarrow \quad \frac{\mathbb{E}[N]}{\pi_{0,0}} = \frac{\mu k(k-1)}{2(\mu-\lambda)} + \frac{\mu\lambda(k\alpha\gamma + \alpha\lambda + \lambda\gamma)}{\alpha\gamma(\mu-\lambda)^2}$$

$$+ \frac{\cancel{\mu\alpha}\lambda\cancel{(\mu-\lambda)}(\mu\lambda - \lambda^2 + k\mu\gamma - k\lambda\gamma - k\gamma^2 - \lambda\gamma)}{\cancel{\alpha}\gamma^2(\mu-\lambda)\cancel{^2}(\mu-\lambda-\gamma)}$$

$$\Rightarrow \quad \frac{\mathbb{E}[N]}{\pi_{0,0}} = \frac{\mu k(k-1)}{2(\mu-\lambda)} + \frac{\mu\lambda(k\alpha\gamma + \alpha\lambda + \lambda\gamma)}{\alpha\gamma(\mu-\lambda)^2} + \frac{\mu\lambda\cancel{(\mu-\lambda-\gamma)}(\lambda+k\gamma)}{\gamma^2(\mu-\lambda)\cancel{(\mu-\lambda-\gamma)}}$$

$$\Rightarrow \quad \mathbb{E}[N] = \frac{k\alpha\gamma(k-1)}{2(k\alpha\gamma + \alpha\lambda + \lambda\gamma)} + \frac{\lambda}{\mu-\lambda} + \frac{\alpha\lambda(\lambda+k\gamma)}{\gamma(k\alpha\gamma + \alpha\lambda + \lambda\gamma)}$$

$$\Rightarrow \quad \mathbb{E}[N] = \mathbb{E}[N^{M/M/1}] + \frac{\alpha\lambda(\lambda+k\gamma)}{\gamma(k\alpha\gamma + \alpha\lambda + \lambda\gamma)} + \frac{k\alpha\gamma(k-1)}{2(k\alpha\gamma + \alpha\lambda + \lambda\gamma)}. \tag{5.42}$$

An application of Little's Law gives the expected response time.

$$\mathbb{E}[R] = \mathbb{E}[R^{M/M/1}] + \frac{1}{\gamma}\frac{k\alpha\gamma + \alpha\lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} + \frac{1}{2\lambda}\frac{k\alpha\gamma(k-1)}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} \tag{5.43}$$

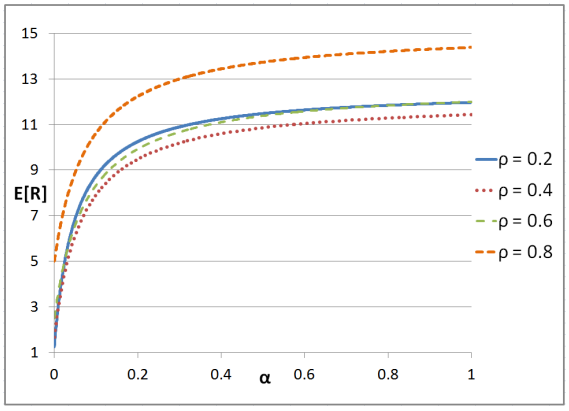
As expected both $\mathbb{E}[N]$ and $\mathbb{E}[R]$ can be written as decompositions involving an $M/M/1$ queue. However, the major difference between the expression for $\mathbb{E}[N]$ and $\mathbb{E}[R]$ in the context of an $M/M/1 \circ \{M, M, 1\}$ queue, is here a third term is present. One should expect that as k increases so should $\mathbb{E}[R]$ (jobs expect to wait while the server is off). While it is true that k is present in the second term, it is also true that this term is bounded above by $\frac{1}{\gamma}$, and therefore an increase in k has a limited impact. On the other hand, the third term of the expression is not bounded in k at all and in fact k is raised to the second power in the numerator and only the first power in the denominator.

Figure 5.5 illustrates the effect which k has on the response time across the range of

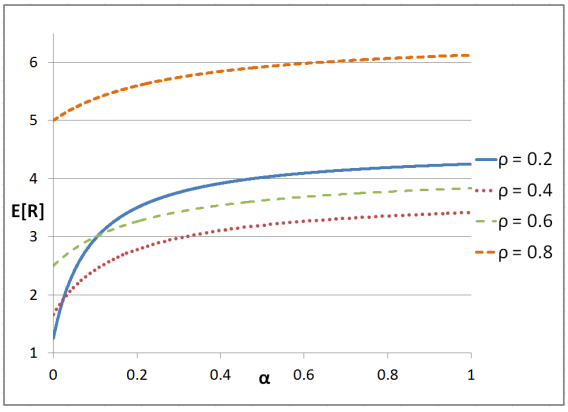
α. Increasing k has little impact on the shape of the curves. However, the distance between the bounds of $\mathbb{E}[R]$ increases with k, and how much it increases seems to rely on the load on the system. For example, comparing Figure 5.5-(a) to Figure 5.5-(c), one can see a large increase in the distance between the bounds for $\rho = 0.2$ ($\approx 50\%$), while for $\rho = 0.8$ the increase is relatively low (less than 20%). Due to this difference of the rate in which the upper bound increases, (perhaps surprisingly) the worst case response time for the lighter loaded systems is greater than that of the heavily loaded systems.

In other words, with respect to response time, it is less appealing to have the server in a lightly loaded system turn off, than it is to have the server in a heavily loaded system turn off, when $k > 0$.

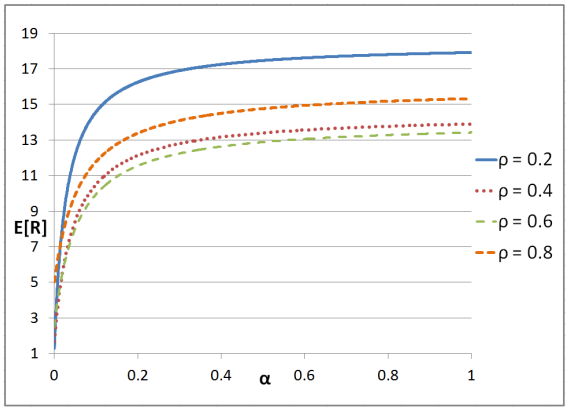
This observation may seem counter intuitive, as one would expect that even for the expected response time, it would be appealing to turn off the server in the system which idles more. While this intuition is valid for an $M/M/1 \circ \{M, M, 1\}$ queue, it can be misleading for an $M/M/1 \circ \{M, M, k\}$ queue. In the case of $k > 0$ there is a phenomenon which occurs in the system, in particular in one where the system is lightly loaded, or one where k is relatively high. If a job arrives while the server is in state *OFF* and there are less than k jobs in the system, that specific job may have to wait a much longer period of time than others which arrive while the server is in states *BUSY* or even state *SETUP*. This is due to the fact that that job has to wait for potentially $k - 1$ jobs to arrive before the server even begins to turn on, and if the system load is light, this takes more time to occur than if the system load were



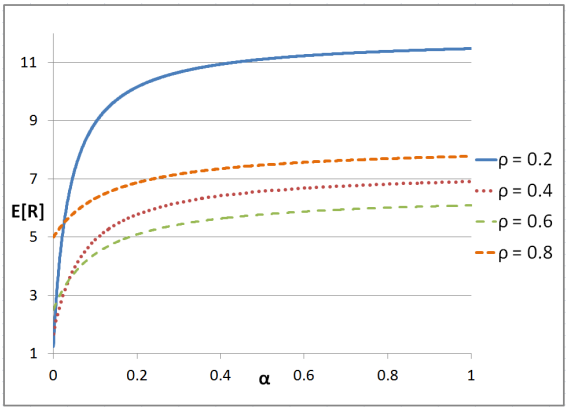
(a) $\mu = 1$, $\gamma = 0.1$, $k = 2$



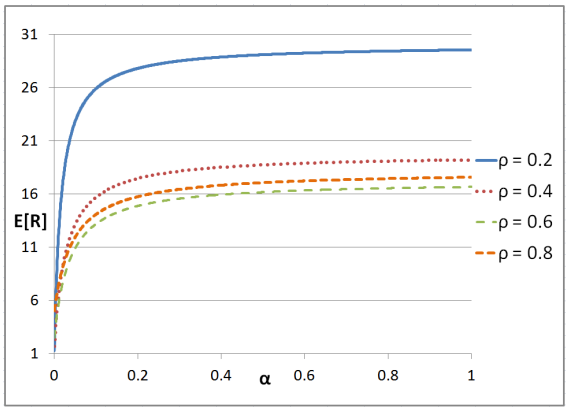
(b) $\mu = 1$, $\gamma = 1$, $k = 2$



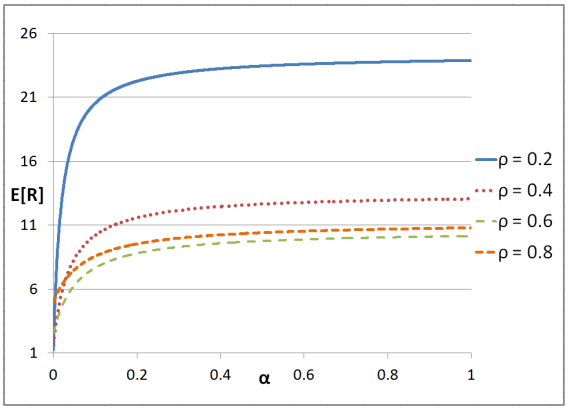
(c) $\mu = 1$, $\gamma = 0.1$, $k = 5$



(d) $\mu = 1$, $\gamma = 1$, $k = 5$



(e) $\mu = 1$, $\gamma = 0.1$, $k = 10$



(f) $\mu = 1$, $\gamma = 1$, $k = 10$

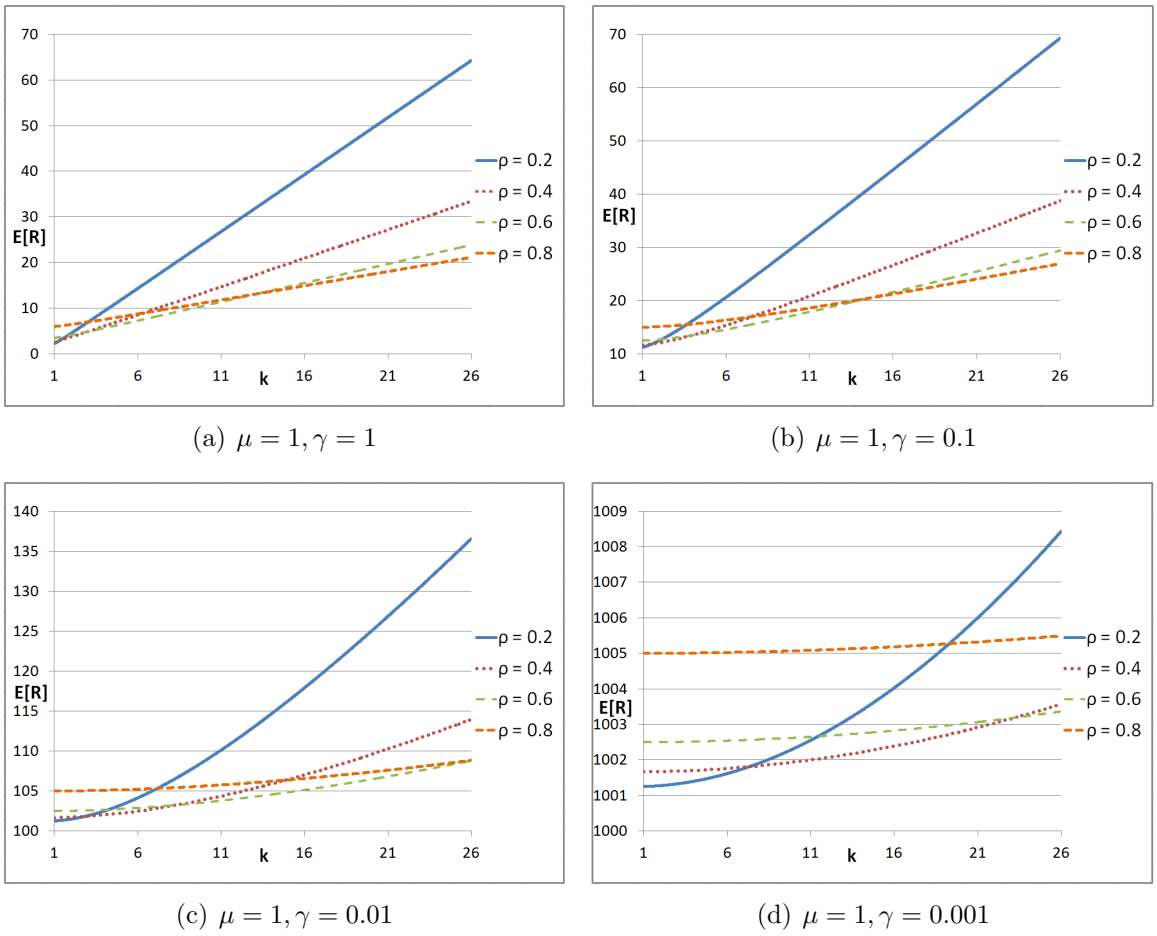
Figure 5.5: $M/M/1 \circ \{M, M, k\}$ response time vs α for varying γ and k values

heavy. Furthermore, if the system load is light, the server will find itself in state *OFF* more often than if the load is heavy (holding all else constant), leading to even more cases of some jobs having to wait a long time in the queue. This behaviour of jobs arriving while the server is turned off skews the mean response time of the system, which explains why Figure 5.5 shows higher response time for lightly loaded systems for certain values of α.

Figure 5.6 takes a closer look at how varying the value of k affects the mean response time. For all sub-figures in Figure 5.6, it is assumed that the server instantly turns off when it idles. This is done since it is known that in the optimal policy this will be the case, or the server will always remain on, and here k would have no impact on the steady state behaviour. Hence, for this context it follows that,

$$\mathbb{E}[R] = \mathbb{E}[R^{M/M/1}] + \frac{1}{\gamma} + \frac{1}{2\lambda}\frac{k(k-1)\gamma}{k\gamma+\lambda}.$$

The reader is reminded that although Figure 5.6 shows k on a continuous range, in practice this parameter must take on discrete values. Here one can see that $\mathbb{E}[R]$ for lightly loaded systems does in fact surpass that of heavily loaded systems for some value of k. The threshold for which the expected response time of a lightly loaded system overtakes the heavily loaded one seems to depend on the value of γ. This can be observed by looking at Figure 5.6-(d), where the server setup time is high. One notes that once $k > 20$, the mean response time of the $\rho = 0.2$ system is larger than any other. However, even for relatively large values of k, the $\rho = 0.8$ system has a larger $\mathbb{E}[R]$ than for both the $\rho = 0.4$ and $\rho = 0.6$ systems. In contrast, looking at the



(a) $\mu = 1, \gamma = 1$ (b) $\mu = 1, \gamma = 0.1$

(c) $\mu = 1, \gamma = 0.01$ (d) $\mu = 1, \gamma = 0.001$

Figure 5.6: $M/M/1 \circ \{M, M, k\}$ response time vs k for varying γ values

higher ranges of k in Figure 5.6-(c) (where the setup time is shorter), the expected response time of the $\rho = 0.4$ system now exceeds that of the $\rho = 0.8$ system. Making the setup times even shorter, one observes that in Figure 5.6-(a) and Figure 5.6-(b), as k increases, the heaviest loaded system offers the lowest expected response time. Furthermore, it is noted that the lower the system load, the more drastic the impact on the response time. For example, for $k = 25$ in Figure 5.6-(a), the expected response time of the $\rho = 0.2$ system is more than double that for $\rho = 0.8$, while the difference between the $\rho = 0.6$ and $\rho = 0.8$ is relatively small ($\approx 10\%$). Taking all of these

observations together one should be very careful when making these design decisions for systems of this nature. Specifically, one should avoid using higher values of k if the system load is not correspondingly heavy, as the increase in $\mathbb{E}[R]$ can be dramatic.

When solving for the expected energy used by the system, the previous method of weighting the steady state probabilities of different energy states (π_{Setup}, π_{Busy}, and π_{Idle}) by the corresponding energy values (E_{Setup}, E_{Busy}, and E_{Idle}) is employed. The observations made in Section 5.1 that $\pi_{Busy} = \rho$, and $\pi_{Idle} = \pi_{1,0}$ still hold, but of course in this context $\pi_{1,0}$ is a slightly different expression due to the addition of k. Therefore, it follows that,

$$\pi_{Idle} = (1-\rho)\frac{\alpha\gamma}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}. \tag{5.44}$$

As before, to solve for π_{Setup}, all terms in the steady-state distribution where the server is turning on ($\pi_{0,n}$ where $n \geq 0$) are summed.

$$\begin{aligned}
& \pi_{Setup} = \sum_{n=k}^{\infty} \pi_{0,n} \\
\Rightarrow \quad & \pi_{Setup} = \pi_{0,0} \sum_{n=k}^{\infty} \left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)} \\
\Rightarrow \quad & \pi_{Setup} = \pi_{0,0} \sum_{n=1}^{\infty} \left(\frac{\lambda}{\lambda+\gamma}\right)^{n} \\
\Rightarrow \quad & \pi_{Setup} = \pi_{0,0} \left(\sum_{n=0}^{\infty} \left(\frac{\lambda}{\lambda+\gamma}\right)^{n} - 1\right) \\
\Rightarrow \quad & \pi_{Setup} = \pi_{0,0} \left(\frac{1}{1-\frac{\lambda}{\lambda+\gamma}} - 1\right) \\
\Rightarrow \quad & \pi_{Setup} = \frac{\lambda}{\gamma}\pi_{0,0}
\end{aligned}$$

$$\Rightarrow \quad \pi_{Setup} = (1-\rho)\frac{\alpha\lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} \tag{5.45}$$

Solving for $\mathbb{E}[E]$ gives,

$$\mathbb{E}[E] = E_{Busy}\pi_{Busy} + E_{Idle}\pi_{Idle} + E_{Setup}\pi_{Setup}.$$

Substituting in $\pi_{Busy} = \rho$, (5.44), and (5.45) allows for one to arrive at a closed form expression.

$$\mathbb{E}[E] = E_{Busy}\rho + E_{Idle}(1-\rho)\frac{\alpha\gamma}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} + E_{Setup}(1-\rho)\frac{\lambda\alpha}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}$$

$$\Rightarrow \quad \mathbb{E}[E] = E_{Busy}\left[\rho + (1-\rho)\frac{\lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}(\gamma r_{Idle} + \alpha r_{Setup})\right]$$

$$\Rightarrow \quad \mathbb{E}[E] = E_{Busy}\left[\rho + (1-\rho)\frac{\cancel{k\alpha\gamma + \alpha\lambda + \lambda\gamma}}{\cancel{k\alpha\gamma + \alpha\lambda + \lambda\gamma}} r_{Idle} + (1-\rho)\frac{\alpha}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}(\lambda r_{Setup} - (\lambda + k\gamma)r_{Idle})\right]$$

$$\Rightarrow \quad \mathbb{E}[E] = \mathbb{E}[E_{M/M/1}] + E_{Busy}(1-\rho)\frac{\alpha}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}(\lambda r_{Setup} - (\lambda + k\gamma)r_{Idle})$$

Using the simplification employed before of factoring out E_{Busy} and solving instead for the normalized expected energy yields:

$$\mathbb{E}[E^N] = \mathbb{E}[E_{M/M/1}] + (1-\rho)\frac{\alpha}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}(\lambda r_{Setup} - (\lambda + k\gamma)r_{Idle}) \tag{5.46}$$

Now all that remains to solve for is the expected switching rate. As before, arriving

at a closed form expression is extremely simple, since by observation $\mathbb{E}[Sw]$ equals the product of α and $\pi_{1,0}$, which implies

$$\mathbb{E}[Sw] = \frac{\alpha\lambda\gamma}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} \tag{5.47}$$

With (5.43), (5.46), and (5.47), all metrics of the cost functions are solved for and optimal policies can be derived. Looking at each closed form expression individually from the view point of the decision variables, one begins to understand why most of the optimal policies remained unknown until now, and the core challenges of the problem. Each metric is minimized when the decision variable is at one of its feasible bounds, unfortunately to minimize each metrics the variables are "pulled" in different directions. This is shown in Table 5.6.

	Optimal Values of	
Metric	α	k
$\mathbb{E}[R]$	0	1
$\mathbb{E}[E]$	0 or $\to \infty$	$\to \infty$
$\mathbb{E}[Sw]$	0	$\to \infty$

Table 5.6: Optimal Parameters of Metrics

It is observed that to minimize $\mathbb{E}[E]$, the optimal choice is $\alpha = 0$ when $r_{idle} < \frac{\lambda}{k\gamma+\lambda} r_{setup}$ and $\alpha \to \infty$ otherwise.

5.2.3 Products of Metrics

As mentioned previously, some popular cost functions in the literature are the products of expectations, for example, the Energy Response Product (ERP), which as its

name suggests is $\mathbb{E}[R]\mathbb{E}[E]$. However, little is known about cost functions which are the expectations of products, for example, $\mathbb{E}[R \cdot E]$. In general $\mathbb{E}[R]\mathbb{E}[E] \neq \mathbb{E}[R \cdot E]$, as these two random variables can be highly dependent. While $\mathbb{E}[R]\mathbb{E}[E]$ is a viable and sensible cost function, $\mathbb{E}[R \cdot E]$ is arguably more accurate in determining the behaviour which the ERP attempts to capture, which in some applications makes the expectation of products a more appealing choice.

This section attempts to gain understanding on how one would solve for metrics which are expectations of products, for example $\mathbb{E}[N \cdot E]$, $\mathbb{E}[W \cdot E]$, and $\mathbb{E}[R \cdot E]$, where W is a random variable denoting a job's waiting time in the queue. One should note that these metrics do not belong to the family of cost functions defined by (4.5), and in general, methods to solve for these metrics are unknown. This section looks to analyse only the metric $\mathbb{E}[N \cdot E]$, while later on in Section 5.3.2 these expectations of products are looked at under a more general scope.

To solve for $\mathbb{E}[N \cdot E]$, the steady state probabilities are weighted by the corresponding number in the system, and the energy values of the corresponding energy state. Due to the assumption that $E_{Off} = 0$, and the fact that while the server is idle, there are 0 jobs in the system, the algebra can be simplified to two sums.

$$\mathbb{E}[N \cdot E] = E_{Setup} \sum_{n=k}^{\infty} n\pi_{0,n} + E_{Busy} \sum_{n=0}^{\infty} n\pi_{1,n}$$

$$\Rightarrow \quad \mathbb{E}[N \cdot E] = E_{Setup} \sum_{n=k}^{\infty} n \left(\frac{\lambda}{\lambda + \gamma} \right)^{n-(k-1)} + E_{Busy} \left[\frac{\lambda}{\alpha} \sum_{n=0}^{\infty} n\rho^n + \frac{\lambda}{\mu - \lambda} \sum_{n=0}^{k-1} n \right.$$

$$- \frac{\lambda}{\mu-\lambda}\sum_{n=0}^{\infty} n\rho^n + \frac{\lambda+\gamma}{\mu-\lambda-\gamma}\sum_{n=k}^{\infty} n\left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)}$$
$$- \quad \frac{\mu\gamma}{(\mu-\lambda)(\mu-\lambda-\gamma)}\sum_{n=k}^{\infty} n\left(\frac{\lambda}{\lambda+\gamma}\right)^{n-(k-1)}\Bigg]$$

Substituting in the corresponding expressions in (5.39), (5.40), and (5.41):

$$\frac{\mathbb{E}[N\cdot E]}{\pi_{0,0}} = E_{Setup}\frac{\lambda(\lambda+k\gamma)}{\gamma^2} + E_{Busy}\Bigg[\frac{\lambda+\gamma}{\mu-\lambda-\gamma}\left(\frac{\lambda(\lambda+k\gamma)}{\gamma^2}\right) + \frac{\mu\lambda}{(\mu-\lambda)^2}\left(\frac{\lambda}{\alpha}-\frac{\lambda}{\mu-\lambda}\right) - \frac{\mu\lambda\gamma(k(\mu-\lambda)+\lambda)}{(\mu-\lambda)^3(\mu-\lambda-\gamma)}\Bigg]$$

$$\Rightarrow \quad \frac{\mathbb{E}[N\cdot E]}{\pi_{0,0}} = E_{Setup}\frac{\lambda(\lambda+k\gamma)}{\gamma^2} + E_{Busy}\frac{\lambda+\gamma}{\mu-\lambda-\gamma}\left(\frac{\lambda(\lambda+k\gamma)}{\gamma^2}\right) + E_{Busy}\Bigg[\frac{\mu\lambda^2(\mu-\lambda-\alpha)}{\alpha(\mu-\lambda)^3} - \frac{\mu\lambda\gamma(k(\mu-\lambda)+\lambda)}{(\mu-\lambda)^2(\mu-\lambda-\gamma)}\Bigg]$$

$$\Rightarrow \quad \frac{\mathbb{E}[N\cdot E]}{\pi_{0,0}} = E_{Setup}\frac{\lambda(\lambda+k\gamma)}{\gamma^2} + E_{Busy}\frac{\lambda+\gamma}{\mu-\lambda-\gamma}\left(\frac{\lambda(\lambda+k\gamma)}{\gamma^2}\right) + E_{Busy}\frac{\mu\lambda(\mu\lambda-\lambda^2-\alpha\lambda-\lambda\gamma-k\alpha\gamma)}{\alpha(\mu-\lambda)^2(\mu-\lambda-\gamma)}$$

$$\Rightarrow \quad \frac{\mathbb{E}[N\cdot E]}{\pi_{0,0}} = E_{Setup}\frac{\lambda(\lambda+k\gamma)}{\gamma^2} + E_{Busy}\frac{\alpha(\lambda+\gamma)(\mu-\lambda)^2(\lambda(\lambda+k\gamma))}{\alpha(\mu-\lambda)^2(\mu-\lambda-\gamma)} + E_{Busy}\frac{\gamma^2\mu\lambda(\mu\lambda-\lambda^2-\alpha\lambda-\lambda\gamma-k\alpha\gamma)}{\alpha(\mu-\lambda)^2(\mu-\lambda-\gamma)}$$

$$\Rightarrow \quad \frac{\mathbb{E}[N\cdot E]}{\pi_{0,0}} = E_{Setup}\frac{\lambda(\lambda+k\gamma)}{\gamma^2} + E_{Busy}\frac{\mu\lambda(k\alpha\gamma+\alpha\lambda+\lambda\gamma)}{\alpha\gamma(\mu-\lambda)^2} + E_{Busy}\frac{(\mu^2\alpha\lambda-2\mu\alpha\lambda^2+\alpha\lambda^3-\mu\alpha\lambda\gamma+\alpha\lambda^2\gamma)(\lambda^2+k\lambda\gamma)}{\alpha\gamma^2(\mu-\lambda^2)(\mu-\lambda-\gamma)}$$

$$\Rightarrow \quad \frac{\mathbb{E}[N \cdot E]}{\pi_{0,0}} = E_{Setup}\frac{\lambda(\lambda + k\gamma)}{\gamma^2} + E_{Busy}\frac{\mu\lambda(k\alpha\gamma + \alpha\lambda + \lambda\gamma)}{\alpha\gamma(\mu - \lambda)^2} + E_{Busy}\frac{\lambda^2(\lambda + k\gamma)}{\gamma^2(\mu - \lambda)}$$

$$\Rightarrow \quad \mathbb{E}[N \cdot E] = E_{Busy}\frac{\lambda}{\mu - \lambda} + E_{Setup}(1 - \rho)\frac{\alpha\lambda(\lambda + k\gamma)}{\gamma(k\alpha\gamma + \alpha\lambda + \lambda\gamma)} + E_{Busy}\rho\frac{\alpha\lambda(\lambda + k\gamma)}{\gamma(k\alpha\gamma + \alpha\lambda + \lambda\gamma)}$$

$$\Rightarrow \quad \mathbb{E}[N \cdot E] = \mathbb{E}[(N \cdot E)_{M/M/1}] + E_{Setup}(1 - \rho)\frac{\alpha\lambda(\lambda + k\gamma)}{\gamma(k\alpha\gamma + \alpha\lambda + \lambda\gamma)} + E_{Busy}\rho\frac{\alpha\lambda(\lambda + k\gamma)}{\gamma(k\alpha\gamma + \alpha\lambda + \lambda\gamma)} \tag{5.48}$$

Taking the derivative with respect to k and setting it equal to 0 allows one to obtain the optimal value of k assuming the server does not always remain on. Without loss of generality, it is assumed $E_{Busy} = 1$.

$$\frac{\partial}{\partial k}\mathbb{E}[N \cdot E] = (\rho + r_{Setup}(1 - \rho))\frac{\alpha\lambda}{\gamma}\left(\frac{\partial}{\partial k}\frac{\lambda + k\gamma}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}\right)$$

$$\Rightarrow \quad \frac{\partial}{\partial k}\mathbb{E}[N \cdot E] = (\rho + r_{Setup}(1 - \rho))\frac{\alpha\lambda}{\gamma}\left(\frac{\gamma(k\alpha\gamma + \alpha\lambda + \lambda\gamma) - \alpha\gamma(\lambda + k\gamma)}{(k\alpha\gamma + \alpha\lambda + \lambda\gamma)^2}\right)$$

$$\Rightarrow \quad \frac{\partial}{\partial k}\mathbb{E}[N \cdot E] = (\rho + r_{Setup}(1 - \rho))\frac{\alpha\lambda}{\gamma}\left(\frac{\lambda\gamma^2}{(k\alpha\gamma + \alpha\lambda + \lambda\gamma)^2}\right) \tag{5.49}$$

Upon inspection one can note that in general (5.49) can not equal 0, for any value of k. This implies that $\mathbb{E}[N \cdot E]$ is minimized when k is at one of its bounds. Since one can also note that (5.49) is always positive ($\mathbb{E}[N \cdot E]$ increases with k), the minimum

must be at the lower bound of k, which is 1 (or possibly 0 if one were to allow such behaviour).

When taking the partial derivative of (5.48) with respect to the second decision variable, α, one sees a similar result.

$$\frac{\partial}{\partial \alpha}\mathbb{E}[N \cdot E] = (\rho + r_{Setup}(1-\rho))\frac{\lambda(\lambda + k\gamma)}{\gamma}\left(\frac{\partial}{\partial k}\frac{\alpha}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}\right)$$

$$\Rightarrow \quad \frac{\partial}{\partial \alpha}\mathbb{E}[N \cdot E] = (\rho + r_{Setup}(1-\rho))\frac{\lambda(\lambda + k\gamma)}{\gamma}\left(\frac{k\alpha\gamma + \alpha\lambda + \lambda\gamma - \alpha(\lambda + k\gamma)}{(k\alpha\gamma + \alpha\lambda + \lambda\gamma)^2}\right)$$

$$\Rightarrow \quad \frac{\partial}{\partial \alpha}\mathbb{E}[N \cdot E] = (\rho + r_{Setup}(1-\rho))\frac{\lambda(\lambda + k\gamma)}{\gamma}\left(\frac{\lambda\gamma}{(k\alpha\gamma + \alpha\lambda + \lambda\gamma)^2}\right) \tag{5.50}$$

Here one can see that (5.50) also cannot equal 0. Furthermore, the derivative is positive, meaning the optimal value for α is at its corresponding lower bound, in this case 0. Taking both these results together, it can be seen that the server has a strong affinity to remain on. In fact, one can conclude that to minimize $\mathbb{E}[N \cdot E]$ under any parameter configuration, it is always optimal to keep the server on. At first thought this may seem like a surprising result. But, after a few observations, one can conclude that this is the case for a large group of metrics of this form, and in fact this result is quite intuitive. However, it turns out that these observations can be made under a more general setting, and therefore are presented later on in Section 5.3.2.

5.3 The M/G/1 ∘ {G,G,k} Queue

Here all assumptions on the underlying distributions of the model are relaxed to the most general case, excluding the arrival stream. This makes the system much more challenging to analyse since it can no longer be viewed as a Markov chain, nor does it contain an embedded Markov chain that aids analysis. Despite these difficulties however, it will be shown that this system can still be analysed with respect to the expected energy used, as well as the expected switching rate.

When dealing with general distributions, it is no longer useful to look at the specific *system states*, (n_1, n_2), denoting the number of jobs in the system, as well as if the server is on or off, such as was done in Sections 5.1 and 5.2. The reason that dividing the model into these states is no longer beneficial is due to the loss of the Markovian, or memoryless property, which is a property of the exponential distribution. If one were to inspect the state $(1, 2)$ for example, it would not be enough to know just this information to make predictions about the future. Specifically in this case, one would also need to keep track of how long the system has been idle since its last turn on, as well as how long the current job has been processed. Instead, the system is viewed from a higher level perspective through its *energy states* of *OFF*, *SETUP*, *BUSY*, and *IDLE* as defined in Section 4.1. Furthermore, a specific state within the state *OFF* is also defined to denote the server being off with no jobs in the system, OFF_0.

5.3.1 The Work-Cycle

Before any analysis of the $M/G/1 \circ \{G, G, k\}$ queue is shown, it is important to first introduce the notion of a *Work-cycle*. A Work-cycle is defined to be the evolution of

the system starting in state OFF_0, leaving, and then returning to OFF_0. In detail, the system starts a Work-cycle with the server being off with no jobs present, k jobs arrive and the server begins to turn on and enters $SETUP$. Once the server turns on, jobs are processed and the server will eventually become idle. The system may switch between $BUSY$ and $IDLE$ an arbitrary number of times, but the system will eventually reach its idle threshold and turn off, returning to OFF_0. This basic concept will allow for an easy analysis of the system with respect to energy used, and the switching rate of the server. To solve for these metrics, some notation must be defined. The proportion of time spent in an energy state in steady state is denoted by P_{IDLE}, P_{SETUP}, P_{BUSY}, P_{OFF}, and $P_{OFF,0}$, respectively. Also, the rate at which Work-cycles occur in the system is denoted by w_{rate}. From here the following observations are made.

- Work-cycles are mutually independent. This comes from the fact that each underlying random variable is independently distributed and when the system reaches state OFF_0 all information present in the system needed to determine future events is completely reset.

- The expected proportion of time spent in any state of the system during one Work-cycle is equal to the proportion of time spent in the corresponding state in steady state. Since the system information is reset when the system enters state OFF_0, this must be the case.

- The system evolution can be viewed as an infinite series of Work-cycles, and the steady state values are equal to the product of the Work-cycle rate and the expected time being in the corresponding state for a single Work-cycle.

The last observation allows one to write out the steady state values for the energy states, in terms of the Work-cycle rate, as seen in (5.51). It is known for a single Work-cycle, the expected time to be in state OFF is k/λ, the expected time to be in state $SETUP$ is $1/\gamma$, and the expected time to be in state OFF is $1/\alpha$.

$$P_{OFF} = \frac{kw_{rate}}{\lambda}, \qquad P_{SETUP} = \frac{w_{rate}}{\gamma}, \qquad P_{IDLE} = \frac{w_{rate}}{\alpha} \tag{5.51}$$

We cannot use the Work-cycle approach for P_{BUSY}. However, P_{BUSY} is actually given for free, due to the well known fact that for any stable single server system the proportion of time the server is busy is equal to the utilization of the server, i.e. $P_{BUSY} = \rho$. As a side note it is very easy, albeit unnecessary, to solve for the expected time the server is in state $BUSY$ for a single Work-cycle, by working backwards:

$$\rho = t_{BUSY} w_{rate} \quad \Rightarrow \quad t_{BUSY} = \frac{\rho}{w_{rate}}$$

With the steady state values for all four of the energy states written with respect to the Work-cycle rate, the fact that all four of the probabilities must sum to 1 may be invoked. From here one can arrive at a closed form expression for the Work-cycle rate, seen in (5.52).

$$1 = P_{OFF} + P_{SETUP} + P_{BUSY} + P_{IDLE}$$

$$\Rightarrow \quad 1 = \frac{kw_{rate}}{\lambda} + \frac{w_{rate}}{\gamma} + \rho + \frac{w_{rate}}{\alpha}$$

$$\Rightarrow \quad 1 - \rho = w_{rate} \frac{k\alpha\gamma + \alpha\lambda + \lambda\gamma}{\alpha\lambda\gamma}$$

$$\Rightarrow \quad w_{rate} = (1 - \rho) \frac{\alpha\lambda\gamma}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} \tag{5.52}$$

From this point one can go on to solve for $\mathbb{E}[E]$, as well as $\mathbb{E}[Sw]$. But before that analysis is shown, an observation is made. It is noted that the Work-cycle rate is intuitively equal to several other values present in the system, including the server's turn on rate, the turn off rate, and the rate out of OFF_0. This is an interesting observation since we get the following result.

$$\lambda P_{OFF,0} = w_{rate} \quad \Rightarrow \quad P_{OFF,0} = (1 - \rho) \frac{\alpha\gamma}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} \quad \Rightarrow \quad P_{OFF,0} = \pi_{0,0},$$

where $\pi_{0,0}$ is the steady state probability from the CTMC in Section 5.2. Furthermore, due to the Poisson process within the state OFF the system is Markovian, implying that for every system state within the energy state OFF, the steady state values of the $M/G/1 \circ \{G, G, k\}$ queue are exactly equal to those of an $M/M/1 \circ \{M, M, k\}$ queue. It will be seen that these systems share other non-trivial characteristics later in this section. For now the derivation of $\mathbb{E}[E]$ is continued.

Firstly each energy state is weighted by the corresponding energy value. Then the value of E_{Busy} is factored out.

$$\mathbb{E}[E] = \rho E_{Busy} + E_{Idle} \frac{\lambda}{\alpha} \pi_{0,0} + E_{Setup} \frac{\lambda}{\gamma} \pi_{0,0}$$

$$\Rightarrow \quad \mathbb{E}[E] = \rho E_{Busy} + E_{Idle}(1-\rho)\frac{\lambda\gamma}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} + E_{Setup}(1-\rho)\frac{\alpha\lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}$$

$$\Rightarrow \quad \mathbb{E}[E] = E_{Busy}\left[\rho + (1-\rho)\frac{\lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}(\gamma r_{Idle} + \alpha r_{Setup})\right]$$

$$\Rightarrow \quad \mathbb{E}[E] = E_{Busy}\left[\rho + (1-\rho)\frac{k\alpha\gamma + \alpha\lambda + \cancel{\lambda\gamma}}{\cancel{k\alpha\gamma} + \alpha\lambda + \lambda\gamma} r_{Idle} + (1-\rho)\frac{\alpha}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}(\lambda r_{Setup} - (\lambda + k\gamma) r_{Idle})\right]$$

$$\Rightarrow \quad \mathbb{E}[E] = \mathbb{E}[E_{M/G/1}] + E_{Busy}(1-\rho)\frac{\alpha}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}(\lambda r_{Setup} - (\lambda + k\gamma) r_{Idle})$$

The typical simplification to normalize by E_{Busy} is applied, arriving at

$$\mathbb{E}[E^N] = \mathbb{E}[E^N_{M/G/1}] + (1-\rho)\frac{\alpha}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}(\lambda r_{Setup} - (\lambda + k\gamma) r_{Idle}). \tag{5.53}$$

The analysis for $\mathbb{E}[Sw]$ is extremely straightforward, only requiring to exploit an observation which was previously made. The turn off rate of the server ($\mathbb{E}[Sw]$) is exactly equal to the Work-cycle rate of the system. It directly follows that,

$$\mathbb{E}[Sw] = w_{rate} \quad \Rightarrow \quad \mathbb{E}[Sw] = (1-\rho)\frac{\alpha\lambda\gamma}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}. \tag{5.54}$$

Again decompositions of the metrics are seen, showing a deep relationship to the non-energy-aware counterpart. However, similarities beyond the previously analysed decomposition are present. Looking at equations (5.53) and (5.54), and comparing them to the equations in Section 5.2 of (5.46) and (5.47), it can be seen that the

mean rate that energy is consumed by the system as well as the switching rate of an $M/G/1 \circ \{G, G, k\}$ system are identical to that of an $M/M/1 \circ \{M, M, k\}$ system. Again, the behaviour of these systems is shown to be completely insensitive to the underlying distributions, save for the arrival stream. Along with this information, we also see the expression

$$\lambda r_{Setup} - (\lambda + k\gamma) r_{Idle}, \tag{5.55}$$

embedded within the second term of (5.53). This term can become negative, and when it does, $\mathbb{E}[E^N]$ is minimized as $\alpha \to \infty$. This has the physical interpretation that when the ratio of energy used while the server is idle is greater than some factor of the ratio of energy used while the server is turning on, it is optimal with respect to the energy used by the system to instantly turn off the server when it starts to idle. The flip side of this situation also exists. If (5.55) is positive, $\mathbb{E}[E^N]$ is minimized when $\alpha = 0$. The physical interpretation here is, when the ratio of energy used while the server is idle is less than some factor of the ratio of energy used while the server is turning on, it is optimal with respect to the energy used by the system to always keep the server on.

The first case of instantly shutting the server off when it idles to minimize energy use makes sense, since it stops the server from idling and brings it to a state which consumes no energy. The second case where energy is minimized when the server always remains on is perhaps not surprising, but is however less intuitive. These observations can be leveraged to yield a very easy way to determine the optimal policy for the system under certain conditions, specifically when $r_{idle} < \frac{\lambda}{k\gamma+\lambda} r_{setup}$. It is obvious that $\mathbb{E}[R]$ and $\mathbb{E}[Sw]$ are both minimized when $\alpha = 0$, so if it is known

that $\mathbb{E}[E^N]$ is also minimized when $\alpha = 0$, it immediately follows that the optimal policy for the system is to simply always leave the server on.

Theorem 3. *The proportion of time spent in the energy states of an $M/G/1 \circ \{G, G, k\}$ queue, is dependent only on the means of the underlying distributions, giving general expressions for $\mathbb{E}[E^N]$ and $\mathbb{E}[Sw]$. That is,*

$$\mathbb{E}[E^N] = \mathbb{E}[E^N_{M/G/1}] + (1-\rho)\frac{\alpha}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}(\lambda r_{Setup} - (\lambda + k\gamma)r_{Idle}),$$
$$\mathbb{E}[Sw] = (1-\rho)\frac{\alpha\lambda\gamma}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}$$

and for any single server system where the arrivals follow a Poisson process, if $r_{idle} < \frac{\lambda}{k\gamma+\lambda} r_{setup}$, it is always optimal to leave the server on.

5.3.2 Products of Metrics

As was seen in Section 5.2.3 the cost function $\mathbb{E}[E \cdot N]$ had the property that it is always optimal to keep the server on. This section explains why this is the case, even under general assumptions of an $M/G/1 \circ \{G, G, k\}$ queue. Furthermore, it will be seen that this is also the case for a broad range of other metrics which are of the form:

$$f(w) = \mathbb{E}[R^{w_1} \cdot E^{w_2} \cdot N^{w_3} \cdot Sw^{w_4} \cdot W^{w_5}], \tag{5.56}$$

where $\forall i. w_i \geq 0$.

With regards to the "always on" property of the $\mathbb{E}[E \cdot N]$ cost function, this becomes intuitively clear after several observations. Firstly, it is known that in a stable system

there is no avoiding being in state $BUSY$ for a proportion of time equal to the system utilization, ρ. Therefore, the energy being consumed by the system must equal E_{Busy} for a proportion ρ of the time. Secondly, it is observed that the expected number in the system while in state $BUSY$ given that it arrived from state $IDLE$, is less than or equal to the expected number of jobs in the system while in state $BUSY$ given that it arrived from state $SETUP$. This is due to the fact that arriving from state $IDLE$ implies there is only one job in the system, while arriving from state $SETUP$ there are at least k jobs, as well as whatever jobs arrived during the setup process (expected to be $\frac{\lambda}{\gamma}$). Thirdly, due to the two previous observations, ignoring the addition of terms to $\mathbb{E}[N \cdot E]$ when the system is in state $IDLE$, OFF, and $SETUP$, one cannot achieve a lower $\mathbb{E}[N \cdot E]$ than the policy which always keeps the server on. Lastly, it is noted that when the system is in state $IDLE$, $N = 0$, which implies $N \cdot E = 0$. Therefore from the third and fourth observation, one can conclude the policy which will always minimize $\mathbb{E}[N \cdot E]$ is the policy which always keeps the server on. This is the exact result which was seen in the algebra in Section 5.2.3.

This same argument can be extended to $\mathbb{E}[W \cdot E]$, since while the system is in state $IDLE$, the waiting time of a job will also always be 0. Furthermore, any cost function of the form (5.56) in which Sw has a non-zero weight, will also be minimized when the server is always on, since in that configuration Sw will equal 0. One begins to see that for a large portion of these cost functions, it is simply optimal to leave the server on.

In fact, for all cost functions of the form (5.56), if $w_3 > 0$, $w_4 > 0$, or $w_5 > 0$, then it

is optimal to have the server remain on. This is because while the server is idle, W and N equal 0. Also, as noted previously if Sw has a non-zero weight, it is trivially optimal to leave the server on. Based on these observations many of the cost functions can be removed from (5.56), as they have trivial solutions. By simplification this gives a new family of cost functions of metric products of the form:

$$f(w) = \mathbb{E}[R^{w_1} \cdot E^{w_2}], \tag{5.57}$$

where $\forall i.w_i \geq 0$.

Observe that $R = W + S$ where S is a random variable denoting the service time of a job. A further observation is made that S is independent from both W, and E. This gives the following equality,

$$\mathbb{E}[S^{w_1} \cdot W^{w_2} \cdot E^{w_3}] = \mathbb{E}[S^{w_1}]\mathbb{E}[W^{w_2} \cdot E^{w_3}]. \tag{5.58}$$

It is observed that $\mathbb{E}[S^{w_1}]$ depends only on the service time distribution, and remains constant no matter what type of policy is chosen. Furthermore, due to reasons discussed earlier, $\mathbb{E}[W^{w_2} \cdot E^{w_3}]$ is minimized when the server always remains on. Substituting $R = W + S$ into (5.57) yields:

$$f(w) = \mathbb{E}[(W + S)^{w_1} \cdot E^{w_2}] \tag{5.59}$$

Restricting the weight, w_1, to be a positive integer allows one to make further comments about these cost functions. Expanding (5.59) gives an expectation of terms, all

of which contain W and E except for one which equals $S^{w_1} \cdot E^{w_2}$. Letting T denote all terms except $S^{w_1} \cdot E^{w_2}$, and substituting into (5.59) gives the following.

$$\begin{aligned} & f(w) = \mathbb{E}[(W+S)^{w_1} \cdot E^{w_2}] \\ \Rightarrow \quad & f(w) = \mathbb{E}\left[\left(\sum_{i=0}^{w_1-1} \binom{w_i}{i} W^{w_1-i} \cdot S^i + S^{w_1}\right) \cdot E^{w_2}\right] \\ \Rightarrow \quad & f(w) = \mathbb{E}[T + S^{w_1} \cdot E^{w_2}] \\ \Rightarrow \quad & f(w) = \mathbb{E}[T] + \mathbb{E}[S^{w_1} \cdot E^{w_2}] \\ \Rightarrow \quad & f(w) = \mathbb{E}[T] + \mathbb{E}[S^{w_1}]\mathbb{E}[E^{w_2}] \end{aligned} \tag{5.60}$$

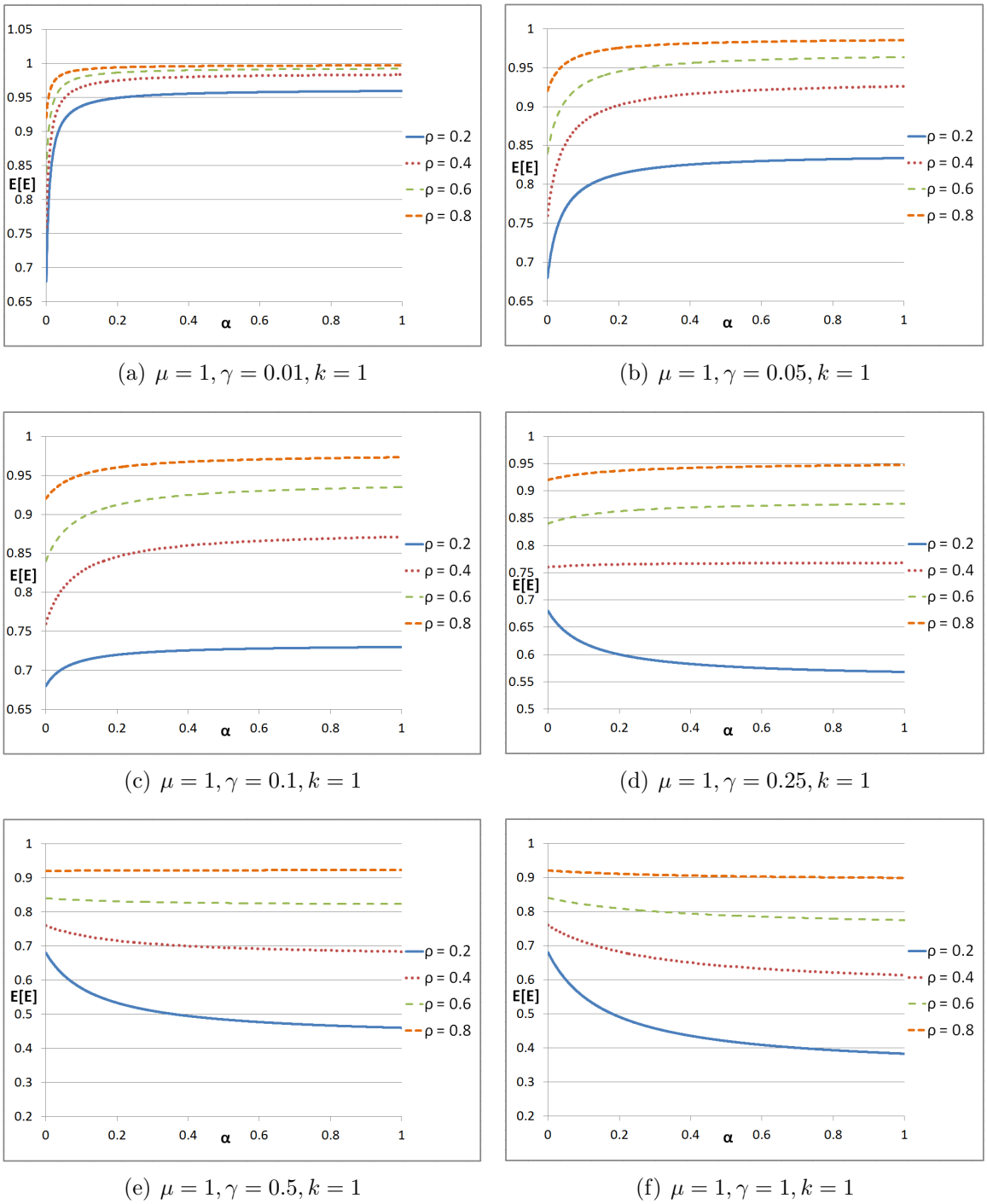
It is noted that all terms contained in T are of the form (5.58) and therefore are minimized when the server remains on. Also, due to the previous observation that $\mathbb{E}[S^{w_1}]$ is independent from the chosen policy, it follows that $\mathbb{E}[S^{w_1}]\mathbb{E}[E^{w_2}]$ is minimized when $\mathbb{E}[E]$ is minimized. It is known from Theorem 3 that if $r_{idle} < \frac{\lambda}{k\gamma+\lambda} r_{setup}$ then $\mathbb{E}[E]$ is also minimized when the server remains on.

From the above observations it is seen that for a large subset of the cost functions which are the expectations of the products, the optimal policies are surprisingly trivial. In fact, for all cost functions of the form (5.56) but not of the form (5.57), it is always optimal to leave the server on. Furthermore, for all cost functions of the form (5.57) if $r_{idle} < \frac{\lambda}{k\gamma+\lambda} r_{setup}$, it is also optimal to always leave the server on.

5.3.3 Energy and Switching

Here the analysis takes a closer look at the effect which different parameter configurations have on $\mathbb{E}[E^N]$ and $\mathbb{E}[Sw]$. Due to the results of Theorem 3 analysing the energy and switching equations (5.53) and (5.54), can be done by only knowing the means of all underlying distributions. This allows one to inspect the expressions of an $M/G/1 \circ \{G, G, k\}$ queue, and make the same observations in the context of an $M/M/1 \circ \{M, M, k\}$ or $M/M/1 \circ \{M, M, 1\}$ queue. Due to this result, conclusions can be made with considerable generality, and is the reason why a more detailed analysis of these metrics has not been presented until now. From this point on, for the purpose of simplicity, it is assumed that $E_{Busy} = 1$, $r_{Idle} = 0.6$, and unless stated otherwise $r_{Setup} = 1$.

The first relationship examined is the effect which the expected turn on time has on the expected energy used by the system. This is shown in Figure 5.7. As one can see, when γ is relatively low, corresponding to longer turn on times (Figures 5.7–(a)-(c)), $\mathbb{E}[E]$ increases with α. This is due to the fact that the term in the expression for the expected energy used, (5.55), is positive in these cases. Furthermore, the lower the value of γ, the larger the increase in $\mathbb{E}[E]$ as α increases. One may argue that analysis of these parameter values is trivial, since due to Theorem 3, one knows that here it would be optimal to have $\alpha = 0$. However it is important to understand the impact α may have on $\mathbb{E}[E]$. For example, a case could arise where due to a mis-estimation of parameters, the server manager instantly turns the server off, but it is actually



(a) $\mu = 1, \gamma = 0.01, k = 1$

(b) $\mu = 1, \gamma = 0.05, k = 1$

(c) $\mu = 1, \gamma = 0.1, k = 1$

(d) $\mu = 1, \gamma = 0.25, k = 1$

(e) $\mu = 1, \gamma = 0.5, k = 1$

(f) $\mu = 1, \gamma = 1, k = 1$

Figure 5.7: $M/G/1 \circ \{G, G, k\}$, $\mathbb{E}[E]$ vs α for varying γ values

optimal to always keep the server on. Understanding these risks can help managers make (or choose to not make) system changes confidently.

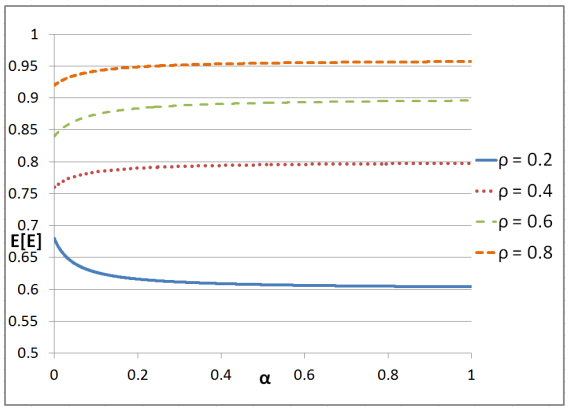
Looking forward at Figures 5.7–(d)-(f), one can begin to see parameter configurations where it is optimal to turn the server off. It is noted that in the case where $\gamma = 0.025$ (Figure 5.7-(d)), $\mathbb{E}[E]$ for a system load of $\rho = 0.2$ decreases with α, while the values of $\mathbb{E}[E]$ for other system loads increase, albeit only slightly. Again this is due to the term (5.55) becoming negative for the lighter load before the other due to the dependence on λ. As γ continues to increase, one can see that $\mathbb{E}[E]$ begins to decrease with α, for all system loads. However, although $\mathbb{E}[E]$ does decrease for all system loads, the heavier the system load is, the less of an impact α tends to have on mean energy used. For example in Figure 5.7-(d), the $\rho = 0.2$ system sees a notable decrease ($\approx 40\%$), while the $\rho = 0.8$ system hardly sees a decrease at all. This is mostly due to the fact that the higher ρ is, the less impact the choice of α can have on the system's $\mathbb{E}[E]$. This is seen intuitively, since one knows that the system must be in state $BUSY$ for a ratio of time equal to the system load, ρ. Furthermore this is seen mathematically, as the α term in (5.53) is weighted by $(1 - \rho)$. In general, the observation is made that the longer the server turn on times are, the more appealing it is from an energy standpoint to keep the server on and the lighter the system load, the more sensitive $\mathbb{E}[E]$ is to changes in γ.

Looking at the relationship between $\mathbb{E}[E]$ and α as k increases tells a similar story, as shown in Figure 5.8. Reaching the threshold at which it becomes advantageous to turn the server instantly off, as opposed to keeping it on, is accelerated by increasing

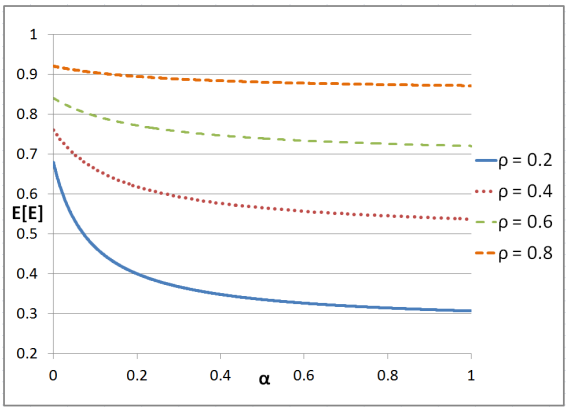
k. This makes sense as having the server turn off and remain in state *OFF* (where the energy consumed is 0) for a longer time is clearly more appealing from an energy standpoint. Furthermore, one notes the same relationship between the system loads as k increases. That is, the percentage difference between having the server turn off, rather than remain on, is much higher when considering systems with relatively lower loads.

This similarity of k and γ is not just seen within the figures. One can also note that in the expression for $\mathbb{E}[E]$, (5.53), as α approaches infinity, k and γ are symmetric. In other words, if the server immediately turns off when it idles, a change to k while holding γ constant is equivalent to making the same change to γ while holding k constant. The interpretation of this is quite interesting, due to the fact that k is directly related to the time the system spends in state *OFF*, while on the other hand, γ is inversely related to the time the system spends in state *SETUP*. The fact that these two parameters have exactly the same relationship to $\mathbb{E}[E]$ is quite remarkable.

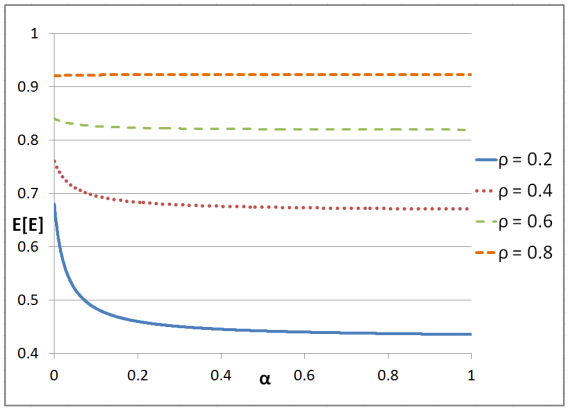
With the effects of k and γ discussed, the analysis proceeds to look at the relationship between $\mathbb{E}[E]$ and r_{Setup}. While one can make the obvious observation that as r_{Setup} increases, so too will the expected energy used, it is of interest to see how exactly it affects the system, specifically with respect to α and ρ. These relationships can be seen in Figure 5.9. As expected, as r_{Setup} increases so does the energy consumption as the server is switched off. However, a result that was perhaps not initially obvious



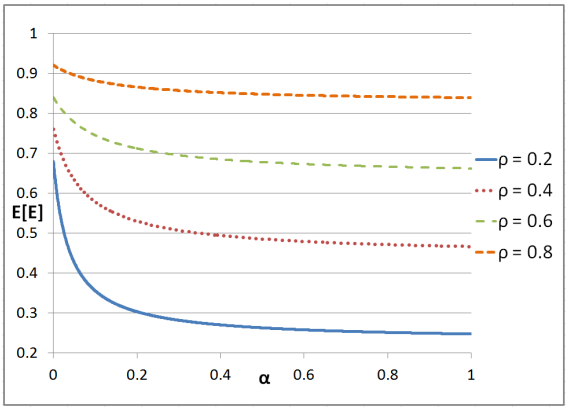
(a) $\mu = 1, \gamma = 0.1, k = 2$



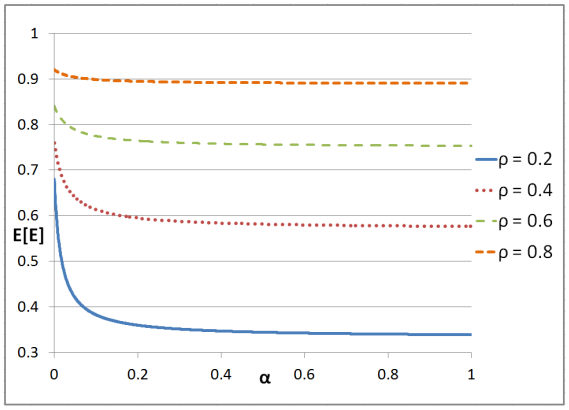
(b) $\mu = 1, \gamma = 1, k = 2$



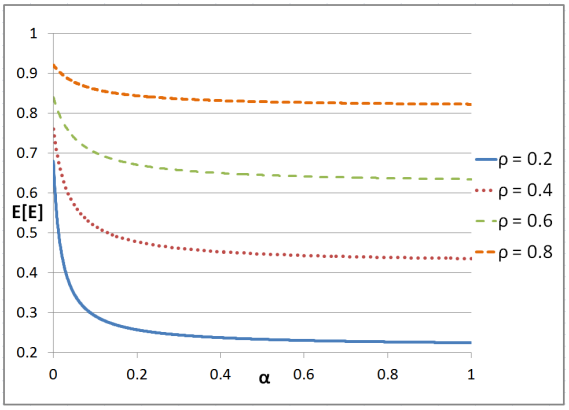
(c) $\mu = 1, \gamma = 0.1, k = 5$



(d) $\mu = 1, \gamma = 1, k = 5$

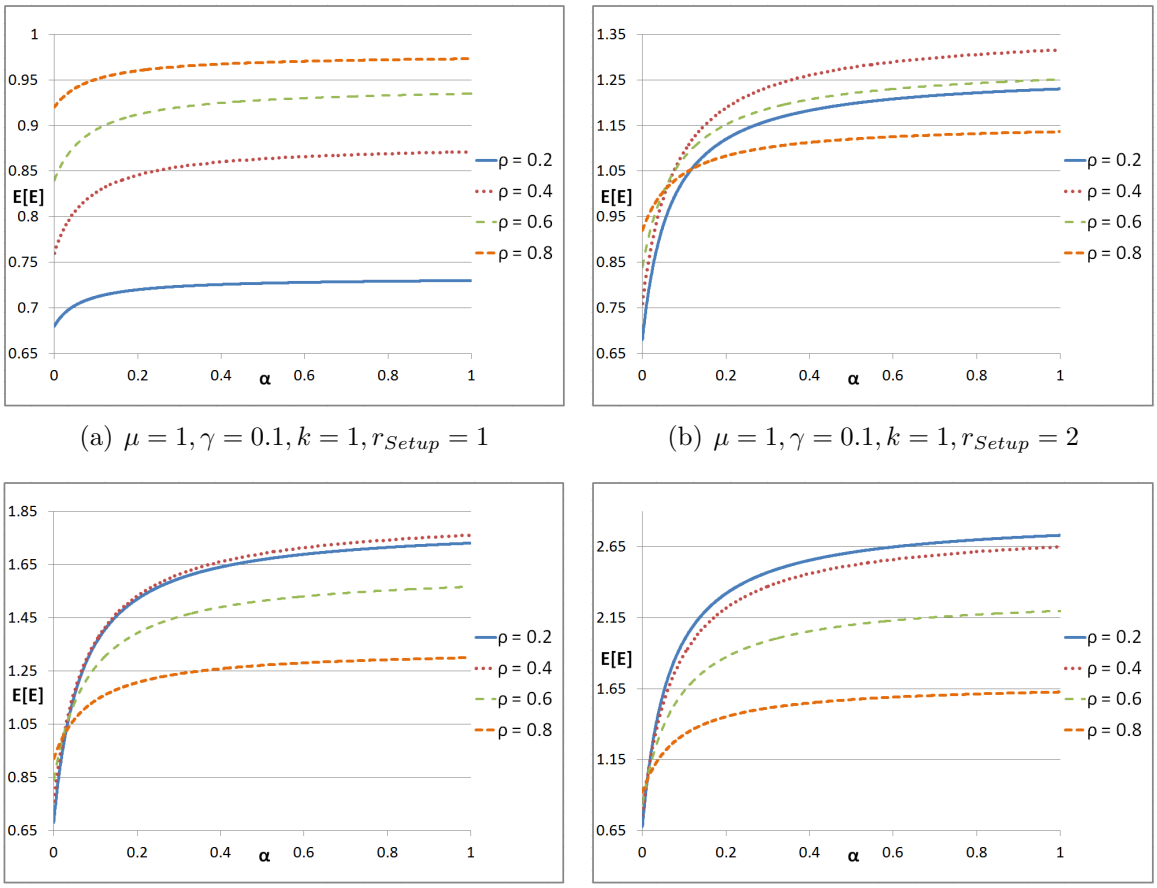


(e) $\mu = 1, \gamma = 0.1, k = 10$



(f) $\mu = 1, \gamma = 1, k = 10$

Figure 5.8: $M/G/1 \circ \{G, G, k\}$, $\mathbb{E}[E]$ vs α for varying k and γ values



(a) $\mu = 1, \gamma = 0.1, k = 1, r_{Setup} = 1$

(b) $\mu = 1, \gamma = 0.1, k = 1, r_{Setup} = 2$

(c) $\mu = 1, \gamma = 0.1, k = 1, r_{Setup} = 3$

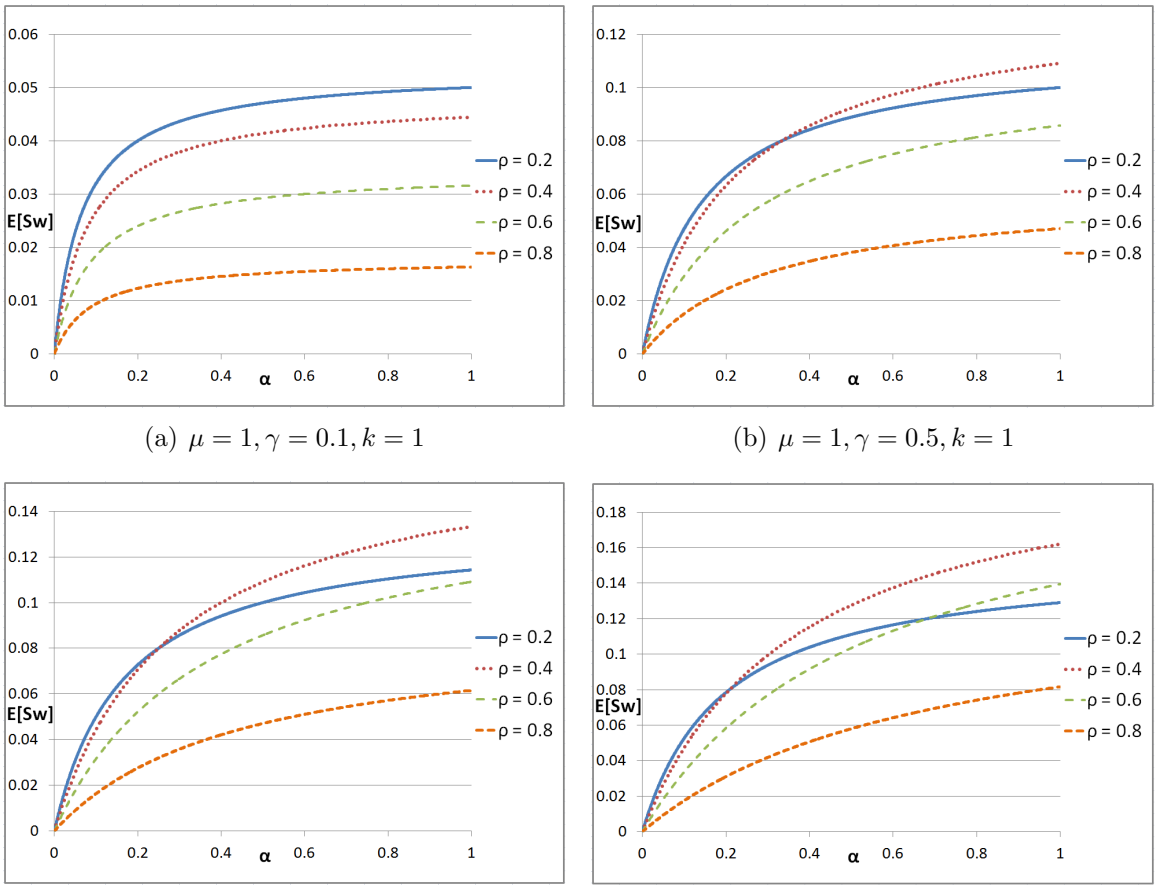
(d) $\mu = 1, \gamma = 0.1, k = 1, r_{Setup} = 5$

Figure 5.9: $M/G/1 \circ \{G, G, k\}$, $\mathbb{E}[E]$ vs α for varying r_{Setup} values

is also noted here. As r_{Setup} increases, $\mathbb{E}[E]$ for lower system loads surpasses that for higher system loads. This is similar to the effect seen with the relationship between $\mathbb{E}[R]$ and k, as shown in Figure 5.6. This is due to the server turning off more frequently as the load decreases. Furthermore, the observation is made that the system loads of $\rho = 0.2$ and $\rho = 0.4$ have little difference between their corresponding values of $\mathbb{E}[E]$ while the gap increases significantly between $\rho = 0.4$, $\rho = 0.6$ and $\rho = 0.8$. Again a similar effect was seen with $\mathbb{E}[R]$ in Figure 5.5. While the curves there were closer when the system loads were heavy, the opposite is seen here with $\mathbb{E}[E]$, as the curves are close when the loads are light.

In conclusion, with regards to $\mathbb{E}[E]$ under general settings, increasing k and γ have the same positive effect on the metric. On the other hand, increasing r_{Setup} will have a negative impact on the overall metric, as well as shifting which system loads have the highest and lowest expectations. Finally, in general, changes to these parameters will have a higher impact for a lightly loaded system, compared to a heavily loaded one.

$\mathbb{E}[Sw]$ is considered next. It is important to note that the expected switching rate is always equal to 0 when the server is kept on at all times ($\alpha = 0$). Furthermore, given this fact, $\mathbb{E}[Sw]$ can also be viewed as a decomposition with the corresponding classical counterpart ($M/G/1$ in this context), as the switching rate for an $M/G/1$ is simply 0.



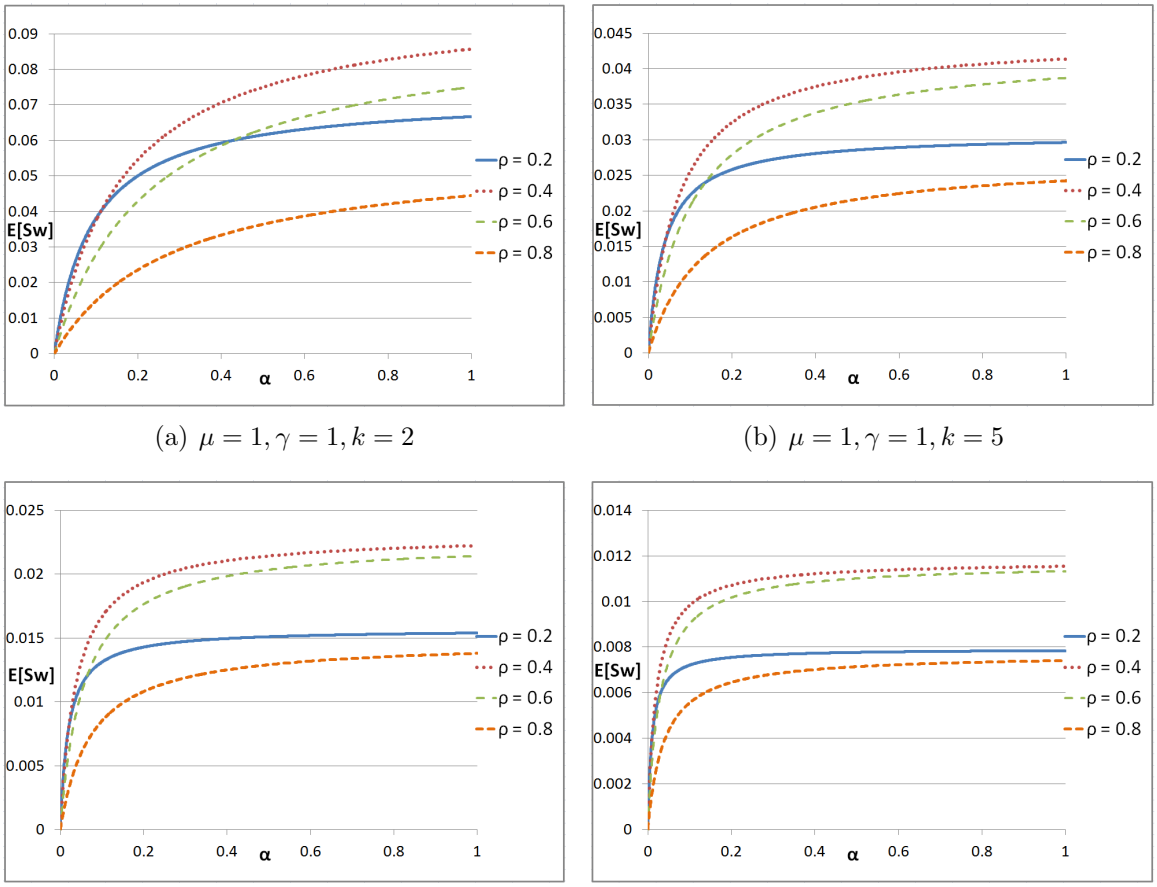
(a) $\mu = 1, \gamma = 0.1, k = 1$ (b) $\mu = 1, \gamma = 0.5, k = 1$

(c) $\mu = 1, \gamma = 1, k = 1$ (d) $\mu = 1, \gamma = 5, k = 1$

Figure 5.10: $M/G/1 \circ \{G, G, k\}$, $\mathbb{E}[Sw]$ vs α for varying γ values

Figure 5.10 shows how increasing γ affects the switching rate for varying system loads. As perhaps initially expected, Figure 5.10-(a) shows the switching rate increase with α, where the lighter the system load, the higher the switching rate. However, when one inspects Figure 5.10–(b)-(d), where the server setup time is increased, one can see that $\mathbb{E}[Sw]$ for the system loads of $\rho = 0.4$ and $\rho = 0.6$ begin to surpass that of the lightest system load $\rho = 0.2$. From this observation, one would perhaps conclude that as γ increases, systems with higher loads will also have higher switching rates. However this is not the case. In fact, allowing γ to become very large (10,000),

gives a similar relationship to that of Figure 5.10-(d). This occurs because there are advantages to having a high or low system load when trying to keep the switching rate low. For a heavily loaded system, the server is less likely be in state *IDLE* and therefore the server does not have an opportunity to turn off. On the other hand, for a lightly loaded system, when the server does shut down and enters state *OFF*, the amount of time for k jobs ($k = 1$ in this context) to arrive, causing the server to turn on, is expected to be higher than for a more heavily loaded system. Systems with medium loads do not benefit from either of these conditions sufficiently often to have



(a) $\mu = 1, \gamma = 1, k = 2$ (b) $\mu = 1, \gamma = 1, k = 5$

(c) $\mu = 1, \gamma = 1, k = 10$ (d) $\mu = 1, \gamma = 1, k = 20$

Figure 5.11: $M/G/1 \circ \{G, G, k\}$, $\mathbb{E}[Sw]$ vs α for varying k values

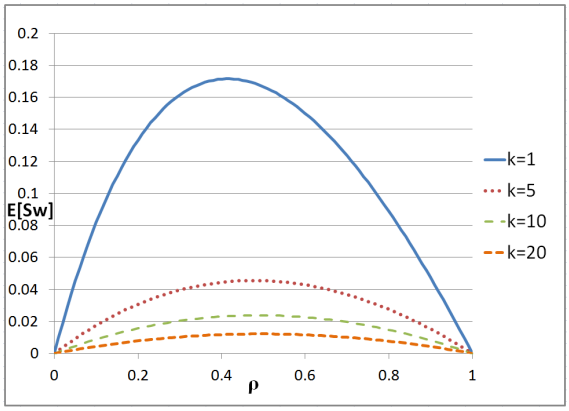


Figure 5.12: $M/G/1 \circ \{G, G, k\}$, $\mathbb{E}[Sw]$ vs ρ, $\mu = 1, \gamma = 1$

their corresponding switching rate be competitive with systems where the load is at one of the extremes.

The effects of this switching rate phenomenon can also be seen in Figure 5.11. However, as one can see, as k and α increase, $\mathbb{E}[Sw]$ for the two medium loaded systems, and the two systems where the loads are at the extremes begin to converge to the same value. This behaviour may initially be viewed as quite odd and unexpected. Looking at the relationship between $\mathbb{E}[Sw]$ and ρ for varying k values, things become clear. Inspecting Figure 5.12, one can see that the expected switching rate has an apparently quadratic relationship to ρ. Furthermore, for higher values of k, these curves become closer to being symmetric around $\rho = 0.5$. For lower values of k, the curves are slightly "slanted" to the left. This explains the convergence of the different system loads as seen before in Figure 5.11, as $\rho = 0.2$, and $\rho = 0.8$, as well as $\rho = 0.4$, and $\rho = 0.6$ all share the same difference from $\rho = 0.5$.

When considering what system configuration to use when concerned about the switching rate, one should understand the relationship that k and ρ have on the metric. The value of γ has relatively little impact, and is often not able to be chosen. On the other hand, increasing k will always decrease $\mathbb{E}[Sw]$. Furthermore, when choosing to have the server shut off, instead of always remaining on, one should note that systems with a medium load will see the greatest increase in the expected switching rate.

5.4 The M/G/1 ∘ {G,M,k} Queue

While one is able to analyse the expected energy and switching costs of an $M/G/1 \circ \{G, G, k\}$ queue, it is much more difficult to arrive at a closed form expression for the expected response time. In order to achieve this goal, the assumption of the idling times being exponentially distributed must again be imposed. The reason for this will be made clear during the analysis presented in this section. However, the reader is reminded that when the arrival stream is a Poisson process (which it is here), the optimal policy will be one which always leaves the server on, or one which instantly turns the server off when it idles. This leads to the fact that the optimal distribution for the idling times is one where the rate is either 0 or ∞. This means that the "shape" of the idling time distribution has no impact on the set of policies which are optimal, but only on the choice of when each of the two policies would be optimal. Furthermore, the assumption that the arrival stream is a Poisson process is typically a reasonable assumption in practice. On the other hand, imposing exponential assumptions on processing and turn on times can be quite inaccurate. Taking all this information together, one can conclude that the $M/G/1 \circ \{G, M, k\}$ queue is a powerful model. Although it is not completely general, it is general where it "counts".

The $M/G/1 \circ \{G, G, k\}$ queue is analysed in a similar way as to the $M/G/1$ queue, with the goal of arriving at a closed form equation for the expected response time of a job, as shown in Section 2.2.4. Firstly, a recursion for the number of jobs in the system is derived, where N_n is a random variable denoting the number of jobs left in the system as the n^{th} job departs.

$$N_{n+1} = \begin{cases} N_n + A_{n+1} - 1 & N_n \geq 1 \\ A_{n+1} & N_n = 0 \end{cases} \tag{5.61}$$

Here, A_{n+1} is a random variable denoting the number of arrivals which occur between the departure of the n^{th} and $(n+1)^{th}$ job, excluding the $(n+1)^{th}$ job itself (if it arrived during that period). In the model, A_{n+1} must also be conditioned on N_n.

$$A_{n+1} = \begin{cases} A_{S,n+1} & N_n \geq 1 \\ A_{S,n+1} + X_{Off,n}(k - 1 + A_{\Gamma,n}) & N_n = 0 \end{cases} \tag{5.62}$$

Here the analysis begins to differ from the classical $M/G/1$ analysis, as more notation is introduced: $A_{S,n}$, $A_{\Gamma,n}$, and $X_{Off,n}$. Firstly, $A_{S,n}$ is a random variable denoting the number of jobs which arrive while the n^{th} job is being processed. Secondly, $A_{\Gamma,n}$ is a random variable denoting the the number of arrivals which occur while the server is turning on after the n^{th} job has left the system. Lastly, $X_{Off,n}$ is an indicator variable that equals 1 when the system is in state *IDLE* after the departure of the n^{th} job, and the next state it moves to is *OFF*, and equals 0 otherwise. It is noted that since all underlying distributions are iid, all of these random variables are independent of

n, and therefore from here on are simply referred to as A_S, A_Γ, and X_{Off}.

Using the Heaviside step function, one can rewrite (5.61) and (5.62) without the use of cases:

$$N_{n+1} = N_n - \mathcal{U}(N_n) + A_{n+1}, \tag{5.63}$$

$$A_{n+1} = A_S + (1 - \mathcal{U}(N_n))X_{Off}(k - 1 + A_\Gamma), \tag{5.64}$$

and after substituting (5.64) into (5.63), one arrives at,

$$N_{n+1} = N_n - \mathcal{U}(N_n) + A_S + (1 - \mathcal{U}(N_n))X_{Off}(k - 1 + A_\Gamma). \tag{5.65}$$

The goal is to arrive at the expected response time. Due to Little's Law, this is equivalent to solving for the expected number of jobs in the system in steady state. To achieve this, one lets $n \to \infty$ and then takes the expectation of both sides of (5.65). However a problem arises when this is done, as $\mathbb{E}[N]$ is present on both sides of the equation and cancels:

$$\cancel{\mathbb{E}[N]} = \cancel{\mathbb{E}[N]} - \mathbb{E}[\mathcal{U}(N)] + \mathbb{E}[A_s] + \mathbb{E}[(1 - \mathcal{U}(N))X_{Off}(k - 1 + A_\Gamma)].$$

The random variables $\mathcal{U}(N)$, X_{Off}, and A_Γ are all independent of each other. This allows the previous equation to be rewritten as,

$$\mathbb{E}[\mathcal{U}(N)] = \mathbb{E}[A_s] + (1 - \mathbb{E}[\mathcal{U}(N)])\mathbb{E}[X_{Off}](k - 1 + \mathbb{E}[A_\Gamma]). \tag{5.66}$$

Although with this equation one cannot reach an expression for $\mathbb{E}[N]$, one can rearrange to solve for $\mathbb{E}[\mathcal{U}(N)]$. At this point in the classical $M/G/1$ analysis it would be seen that $\mathbb{E}[\mathcal{U}(N)] = \rho$. This of course makes perfect sense since the interpretation of $\mathbb{E}[\mathcal{U}(N)]$ is the steady state probability that there is at least one job in the system. In the case of an $M/G/1$ queue, this is equivalent to the server being busy, which is known to be ρ. However, in the analysis of the $M/G/1 \circ \{G, G, k\}$ queue one should not expect $\mathbb{E}[\mathcal{U}(N)]$ to equal ρ, because ρ is not equal to the probability that the system has at least one job in steady state. From Section 5.3 it is known that this probability is given by,

$$P[N > 0] = 1 - P_{Off,0} - P_{Idle}$$

$$\Rightarrow \quad P[N > 0] = 1 - (1 - \rho)\frac{\alpha\gamma}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} - (1 - \rho)\frac{\lambda\gamma}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}$$

$$\Rightarrow \quad P[N > 0] = \frac{1}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}\left(\frac{\mu((k-1)\alpha\gamma + \alpha\lambda) + \lambda(\alpha\gamma + \lambda\gamma)}{\mu}\right). \tag{5.67}$$

As a sanity check, one can rearrange (5.66) and solve for $\mathbb{E}[\mathcal{U}(N)]$ to ensure it equals (5.67). However, before doing so, the expectations of A_S, X_{Off}, and A_Γ must be evaluated. Firstly, $\mathbb{E}[A_S]$ is the product of the arrival rate and the expected time to process a job, which is $\frac{\lambda}{\mu} = \rho$. Secondly, $\mathbb{E}[X_{Off}]$ equals the probability that the system turns off before a job arrives, once the system enters *IDLE*. Due to the memoryless nature of both the idling times and the arrival stream this is easily seen to be $\frac{\alpha}{\alpha+\lambda}$. Here it becomes apparent why the idling times must be exponentially distributed, as otherwise solving for this probability could become very difficult. The

reader is reminded that the system keeps track of how long the server has been idle since the last time it turned on, and not simply how long it has been idle since it entered the state *IDLE*. Lastly $\mathbb{E}[A_\Gamma]$, similar to $\mathbb{E}[A_S]$, is the product of the arrival rate and the expected time it takes to turn on, $\frac{\lambda}{\gamma}$. Putting it all together,

$$\mathbb{E}[A_S] = \rho, \quad \mathbb{E}[X_{Off}] = \frac{\alpha}{\lambda + \alpha}, \quad \text{and} \quad \mathbb{E}[A_\Gamma] = \frac{\lambda}{\gamma}.$$

With these expectations solved, one can now calculate $\mathbb{E}[\mathcal{U}(N)]$.

$$\mathbb{E}[\mathcal{U}(N)] = \frac{\mathbb{E}[A_S] + \mathbb{E}[X_{Off}](k - 1 + \mathbb{E}[A_\Gamma])}{1 + \mathbb{E}[X_{Off}](k - 1 + \mathbb{E}[A_\Gamma])}$$

$$\Rightarrow \quad \mathbb{E}[\mathcal{U}(N)] = \frac{\rho + \dfrac{\alpha}{\alpha + \gamma}\left((k-1) + \dfrac{\lambda}{\gamma}\right)}{\dfrac{\alpha}{\alpha + \gamma}\left((k-1) + \dfrac{\lambda}{\gamma}\right)}$$

$$\Rightarrow \quad \mathbb{E}[\mathcal{U}(N)] = \frac{\dfrac{\lambda\gamma(\alpha + \lambda) + \mu\alpha((k-1)\gamma + \lambda)}{\cancel{\mu\gamma(\alpha + \lambda)}}}{\dfrac{\mu(\lambda\gamma + \alpha\gamma + (k-1)\alpha\gamma + \alpha\lambda)}{\cancel{\mu\gamma(\alpha + \lambda)}}}$$

$$\Rightarrow \quad \mathbb{E}[\mathcal{U}(N)] = \frac{1}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}\left(\frac{\mu((k-1)\alpha\gamma + \alpha\lambda) + \lambda(\alpha\gamma + \lambda\gamma)}{\mu}\right) \tag{5.68}$$

$$\Rightarrow \quad \mathbb{E}[\mathcal{U}(N)] = \rho\frac{\alpha\gamma + \lambda\gamma}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} + \frac{(k-1)\alpha\gamma + \alpha\lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}$$

$$\Rightarrow \quad \mathbb{E}[\mathcal{U}(N)] = \rho + (1-\rho)\alpha\frac{(k-1)\gamma + \lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} \tag{5.69}$$

The sanity check succeeds in showing that $\mathbb{E}[\mathcal{U}(N)]$ is equivalent to $P[N > 0]$, by showing that (5.67) equals (5.68). It also goes further to show that $\mathbb{E}[\mathcal{U}(N)]$ can also be seen as a decomposition, seen in (5.69). Although this result is refreshing to see, as well as building confidence in the analysis, it still remains that one cannot solve for $\mathbb{E}[N]$.

The analysis returns to (5.65), again letting $n \to \infty$, but before taking expectations, both sides of the equation are squared.

$$\begin{aligned} N^2 = N^2 &- 2N\mathcal{U}(N) + 2NA_S + N(1-\mathcal{U}(N))X_{Off}(k-1+A_\Gamma) \\ &+ \mathcal{U}^2(N) - 2\mathcal{U}(N)A_S - 2\mathcal{U}(N)(1-\mathcal{U}(N))X_{Off}(k-1+A_\Gamma) \\ &+ A_S^2 + 2A_S(1-\mathcal{U}(N))X_{Off}(k-1+A_\Gamma) \\ &+ (1-\mathcal{U}(N))^2X_{Off}^2(k-1+A_\Gamma)^2 \end{aligned}$$

Taking expectations of both sides of the previous equation yields,

$$\begin{aligned} \cancel{\mathbb{E}[N^2]} = \cancel{\mathbb{E}[N^2]} &- 2\mathbb{E}[N\mathcal{U}(N)] + 2\mathbb{E}[NA_S] + \mathbb{E}[N(1-\mathcal{U}(N))X_{Off}(k-1+A_\Gamma)] \\ &+ \mathbb{E}[\mathcal{U}^2(N)] - 2\mathbb{E}[\mathcal{U}(N)A_S] - 2\mathbb{E}[\mathcal{U}(N)(1-\mathcal{U}(N))X_{Off}(k-1+A_\Gamma)] \\ &+ \mathbb{E}[A_S^2] + 2\mathbb{E}[A_S(1-\mathcal{U}(N))X_{Off}(k-1+A_\Gamma)] \\ &+ \mathbb{E}[(1-\mathcal{U}(N))^2X_{Off}^2(k-1+A_\Gamma)^2]. \end{aligned} \tag{5.70}$$

At first glance this equation looks daunting, with several expectations which seem

difficult to compute. However some fortunate simplifications can be made due to the following observed equalities, which exploit independence and the definition of the Heaviside step function.

$$\begin{aligned}
\mathbb{E}[NA_S] &= \mathbb{E}[N]\mathbb{E}[A_S] \\
\mathbb{E}[A_S A_\Gamma] &= \mathbb{E}[A_S]\mathbb{E}[A_\Gamma] \\
\mathbb{E}[A_S X_{Off}] &= \mathbb{E}[A_S]\mathbb{E}[X_{Off}] \\
\mathbb{E}[N\mathcal{U}(N)] &= \mathbb{E}[N] \\
\mathbb{E}[\mathcal{U}^2(N)] &= \mathbb{E}[\mathcal{U}(N)] \\
\mathbb{E}[(1-\mathcal{U}(N))^2] &= \mathbb{E}[1-\mathcal{U}(N)] \\
\mathbb{E}[X_{Off}^2] &= \mathbb{E}[X_{Off}] \\
\mathbb{E}[\mathcal{U}(N)(1-\mathcal{U}(N))] &= 0 \\
\mathbb{E}[N(1-\mathcal{U}(N))] &= 0
\end{aligned}$$

Applying these equalities to (5.70) makes the expression much simpler.

$$\begin{aligned}
2\mathbb{E}[N] = 2\mathbb{E}[N]\mathbb{E}[A_S] &+ \mathbb{E}[\cancelto{0}{N(1-\mathcal{U}(N))}X_{Off}(k-1+A_\Gamma)] \\
&+ \mathbb{E}[\mathcal{U}(N)] - 2\mathbb{E}[\mathcal{U}(N)]\mathbb{E}[A_S] \\
&- 2\mathbb{E}[\cancelto{0}{\mathcal{U}(N)(1-\mathcal{U}(N))}X_{Off}(k-1+A_\Gamma)] \\
&+ \mathbb{E}[A_S^2] + 2\mathbb{E}[A_S](1-\mathbb{E}[\mathcal{U}(N)])\mathbb{E}[X_{Off}](k-1+\mathbb{E}[A_\Gamma]) \\
&+ (1-\mathbb{E}[\mathcal{U}(N)])\mathbb{E}[X_{Off}]\mathbb{E}[(k-1+A_\Gamma)^2]
\end{aligned}$$

$$\Rightarrow \quad 2\mathbb{E}[N] = 2\mathbb{E}[N]\mathbb{E}[A_S] + \mathbb{E}[\mathcal{U}(N)] - 2\mathbb{E}[\mathcal{U}(N)]\mathbb{E}[A_S]$$

$$+ \mathbb{E}[A_S^2] + 2\mathbb{E}[A_S](1 - \mathbb{E}[\mathcal{U}(N)])\mathbb{E}[X_{Off}](k - 1 + \mathbb{E}[A_\Gamma])$$
$$+ (1 - \mathbb{E}[\mathcal{U}(N)])\mathbb{E}[X_{Off}]((k-1)^2 + 2(k-1)\mathbb{E}[A_\Gamma] + \mathbb{E}[A_\Gamma^2])$$

Substituting in the values for $\mathbb{E}[A_S]$, $\mathbb{E}[A_\Gamma]$, and $\mathbb{E}[X_{Off}]$ previously derived, and then substituting in (5.69) yields,

$$\begin{aligned}
2\mathbb{E}[N](1-\rho) &= \mathbb{E}[A_S^2] + \mathbb{E}[\mathcal{U}(N)] - 2\rho\mathbb{E}[\mathcal{U}(N)] \\
&\quad + 2\rho(1 - \mathbb{E}[\mathcal{U}(N)])\frac{\alpha}{\alpha+\lambda}\left(k - 1 + \frac{\lambda}{\gamma}\right) \\
&\quad + (1 - \mathbb{E}[\mathcal{U}(N)])\frac{\alpha}{\alpha+\lambda}\left((k-1)^2 + 2(k-1)\frac{\lambda}{\gamma} + \mathbb{E}[A_\Gamma^2]\right)
\end{aligned}$$

$$\begin{aligned}
\Rightarrow \quad 2\mathbb{E}[N](1-\rho) &= \mathbb{E}[A_S^2] + \rho + (1-\rho)\alpha\frac{(k-1)\gamma + \lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} \\
&\quad - 2\rho\left(\rho + (1-\rho)\alpha\frac{(k-1)\gamma + \lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}\right) \\
&\quad + 2\rho\left(1 - \rho - (1-\rho)\alpha\frac{(k-1)\gamma + \lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}\right)\frac{\alpha}{\alpha+\lambda}\left(k - 1 + \frac{\lambda}{\gamma}\right) \\
&\quad + \left(1 - \rho - (1-\rho)\alpha\frac{(k-1)\gamma + \lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}\right)\frac{\alpha}{\alpha+\lambda} \\
&\quad \cdot \left((k-1)^2 + 2(k-1)\frac{\lambda}{\gamma} + \mathbb{E}[A_\Gamma^2]\right)
\end{aligned}$$

$$\begin{aligned}
\Rightarrow \quad 2(1-\rho)\mathbb{E}[N] &= \rho - 2\rho^2 + \mathbb{E}[A_S^2] + (1-\rho)\alpha\frac{(k-1)\gamma + \lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} \\
&\quad - 2\rho(1-\rho)\alpha\frac{(k-1)\gamma + \lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} \\
&\quad + 2\rho(1-\rho)\left(1 - \alpha\frac{(k-1)\gamma + \lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}\right)\frac{\alpha}{\alpha+\lambda}\left(k - 1 + \frac{\lambda}{\gamma}\right) \\
&\quad + (1-\rho)\left(1 - \alpha\frac{(k-1)\gamma + \lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}\right)\frac{\alpha}{\alpha+\lambda}
\end{aligned}$$

$$\cdot \left((k-1)^2 + 2(k-1)\frac{\lambda}{\gamma} + \mathbb{E}[A_\Gamma^2] \right) \tag{5.71}$$

From [12] it is noted that $\mathbb{E}[A_S^2] = \rho + \lambda^2\sigma_S^2$, where σ_S^2 denotes the variance of the service time distribution. Due to this observation, one can make the same argument with respect to $\mathbb{E}[A_\Gamma^2]$, concluding that $\mathbb{E}[A_\Gamma^2] = \frac{\lambda}{\gamma} + \lambda^2\sigma_\Gamma^2$, where σ_Γ^2 denotes the variance of the setup time distribution. For the sake of simplicity, a place-holder variable is defined to keep the algebra clean.

$$\begin{aligned}
& \Gamma = (k-1)^2 + 2(k-1)\frac{\lambda}{\gamma} + \mathbb{E}[A_\Gamma^2] \\
\Rightarrow \quad & \Gamma = (k-1)^2 + 2(k-1)\frac{\lambda}{\gamma} + \frac{\lambda}{\gamma} + \lambda^2\sigma_\Gamma^2 \\
\Rightarrow \quad & \Gamma = (k-1)^2 + (2k-1)\frac{\lambda}{\gamma} + \lambda^2\sigma_\Gamma^2
\end{aligned}$$

Substituting Γ and $\mathbb{E}[A_S^2] = \rho + \lambda^2\sigma_S^2$ into (5.71) gives,

$$\begin{aligned}
\mathbb{E}[N] = \rho &+ \frac{\rho^2 + \lambda^2\sigma_S^2}{2(1-\rho)} + \frac{\alpha}{2}\frac{(k-1)\gamma + \lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} - \rho\alpha\frac{(k-1)\gamma + \lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} \\
&+ \rho\left(1 - \alpha\frac{(k-1)\gamma + \lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}\right)\frac{\alpha}{\alpha + \lambda}\left(\frac{(k-1)\gamma + \lambda}{\gamma}\right) \\
&+ \frac{1}{2}\left(1 - \alpha\frac{(k-1)\gamma + \lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}\right)\frac{\alpha}{\alpha + \lambda}\Gamma
\end{aligned}$$

$$\begin{aligned}
\Rightarrow \quad \mathbb{E}[N] = \mathbb{E}[N_{M/G/1}] &+ \frac{\alpha}{\alpha + \lambda}\left[\rho\frac{(k-1)\gamma + \lambda}{\gamma} + \frac{1}{2}\Gamma\right] \\
&+ \alpha\frac{(k-1)\gamma + \lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}\left[\frac{1}{2} - \rho - \rho\frac{\alpha}{\alpha + \lambda}\left(\frac{(k-1)\gamma + \lambda}{\gamma}\right) - \frac{1}{2}\frac{\alpha}{\alpha + \lambda}\Gamma\right].
\end{aligned} \tag{5.72}$$

Although (5.72) is not as simple as other expressions seen in this chapter, it is still tractable, and the decomposition involving the $M/G/1$ counterpart is clearly seen. It is noted that $\mathbb{E}[N]$ depends on both the mean and variance of the general distributions, but not on any higher moments. The usual application of Little's Law is applied to arrive at $\mathbb{E}[R]$.

$$\begin{aligned}\mathbb{E}[R] = \mathbb{E}[R_{M/G/1}] &+ \frac{\alpha}{\alpha+\lambda}\left[\frac{1}{\mu}\frac{(k-1)\gamma+\lambda}{\gamma} + \frac{1}{2\lambda}\Gamma\right] \\ &+ \alpha\frac{(k-1)\gamma+\lambda}{k\alpha\gamma+\alpha\lambda+\lambda\gamma}\left[\frac{1}{2\lambda} - \frac{1}{\mu} - \frac{1}{\mu}\frac{\alpha}{\alpha+\lambda}\left(\frac{(k-1)\gamma+\lambda}{\gamma}\right) - \frac{1}{2\lambda}\frac{\alpha}{\alpha+\lambda}\Gamma\right]\end{aligned} \tag{5.73}$$

Theorem 4. *The expected number of jobs in, as well as the expected response time of a job in an $M/G/1 \circ \{G, M, k\}$ queue in steady state are dependent only on the first moments of all underlying distributions, as well as the second moments of the service and setup time distributions, and are given by,*

$$\begin{aligned}\mathbb{E}[N] = \mathbb{E}[N_{M/G/1}] &+ \frac{\alpha}{\alpha+\lambda}\left[\rho\frac{(k-1)\gamma+\lambda}{\gamma} + \frac{1}{2}\Gamma\right] \\ &+ \alpha\frac{(k-1)\gamma+\lambda}{k\alpha\gamma+\alpha\lambda+\lambda\gamma}\left[\frac{1}{2} - \rho - \rho\frac{\alpha}{\alpha+\lambda}\left(\frac{(k-1)\gamma+\lambda}{\gamma}\right) - \frac{1}{2}\frac{\alpha}{\alpha+\lambda}\Gamma\right]\end{aligned}$$

and,

$$\begin{aligned}\mathbb{E}[R] = \mathbb{E}[R_{M/G/1}] &+ \frac{\alpha}{\alpha+\lambda}\left[\frac{1}{\mu}\frac{(k-1)\gamma+\lambda}{\gamma} + \frac{1}{2\lambda}\Gamma\right] \\ &+ \alpha\frac{(k-1)\gamma+\lambda}{k\alpha\gamma+\alpha\lambda+\lambda\gamma}\left[\frac{1}{2\lambda} - \frac{1}{\mu} - \frac{1}{\mu}\frac{\alpha}{\alpha+\lambda}\left(\frac{(k-1)\gamma+\lambda}{\gamma}\right) - \frac{1}{2\lambda}\frac{\alpha}{\alpha+\lambda}\Gamma\right],\end{aligned}$$

where

$$\Gamma = (k-1)^2 + (2k-1)\frac{\lambda}{\gamma} + \lambda^2\sigma_\Gamma^2.$$

Taking Theorems 3 and 4 together, one has closed form expressions for $\mathbb{E}[R]$, $\mathbb{E}[E^N]$, and $\mathbb{E}[Sw]$ for an $M/G/1\circ\{G, M, k\}$ queue. With these expressions one can construct any cost function described by (4.5). Once the cost function has been constructed, values of k and α which minimize it can be derived.

Chapter 6

Applications

With the model presented and analysed, the reader is aware that one may derive the optimal policy where the cost function is of the form (4.5), using the material in Chapter 5. This chapter focuses on some of the ways the previous work can be employed to arrive at these policies, of which some are in contexts not yet considered. Implications of past results are also examined here, primarily in the setting of multi-server system with random routing.

6.1 Optimal Parameter Values

One of the more popular cost functions used in the literature is $\mathbb{E}[R] + \beta_1\mathbb{E}[E] + \beta_2\mathbb{E}[Sw]$ [3, 16, 20, 22], and from here on is denoted by $\mathcal{C}$. However, as previously mentioned, the optimal policy for this cost function was not yet known. For this reason, this section focuses on deriving the optimal policy for the weighted sum of the three metrics. This derivation is done under the assumptions of the $M/M/1 \circ \{M, M, k\}$ queue, purely for reasons of simplicity, although the results can be easily extended to

that of an $M/G/1 \circ \{G, M, k\}$ queue.

To determine the optimal policy is to minimize the cost function. This is done by taking the partial derivatives with respect to the decision variables, and setting them equal to 0. The derivation begins by taking the derivative of the cost function with respect to α.

$$\frac{\partial}{\partial\alpha}\mathcal{C} = \frac{\partial}{\partial\alpha}\mathbb{E}[R] + \beta_1\frac{\partial}{\partial\alpha}\mathbb{E}[E^N] + \beta_2\frac{\partial}{\partial\alpha}\mathbb{E}[Sw] \tag{6.74}$$

To keep the algebra clean, each partial derivative is solved individually. Firstly, the expected response time partial derivative with respect to α is derived.

$$\frac{\partial}{\partial\alpha}\mathbb{E}[R] = \frac{\partial}{\partial\alpha}\left(\frac{1}{\gamma}\frac{k\alpha\gamma + \alpha\lambda}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} + \frac{1}{2\lambda}\frac{k\alpha\gamma(k-1)}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}\right)$$

$$\Rightarrow \quad \frac{\partial}{\partial\alpha}\mathbb{E}[R] = \frac{k\gamma + \lambda}{\gamma}\frac{\partial}{\partial\alpha}\frac{\alpha}{k\alpha\gamma + \alpha\lambda + \lambda\gamma} + \frac{k(k-1)\gamma}{2\lambda}\frac{\partial}{\partial\alpha}\frac{\alpha}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}$$

$$\Rightarrow \quad \frac{\partial}{\partial\alpha}\mathbb{E}[R] = \left(\frac{k\gamma + \lambda}{\gamma} + \frac{k(k-1)\gamma}{2\lambda}\right)\frac{\partial}{\partial\alpha}\frac{\alpha}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}$$

$$\Rightarrow \quad \frac{\partial}{\partial\alpha}\mathbb{E}[R] = \left(\frac{k\gamma + \lambda}{\gamma} + \frac{k(k-1)\gamma}{2\lambda}\right)\frac{\cancel{k\alpha\gamma} + \cancel{\alpha\lambda} + \lambda\gamma - \cancel{k\alpha\gamma} - \cancel{\alpha\lambda}}{(k\alpha\gamma + \alpha\lambda + \lambda\gamma)^2}$$

$$\Rightarrow \quad \frac{\partial}{\partial\alpha}\mathbb{E}[R] = \left(\frac{k\gamma + \lambda}{\gamma} + \frac{k(k-1)\gamma}{2\lambda}\right)\frac{\lambda\gamma}{(k\alpha\gamma + \alpha\lambda + \lambda\gamma)^2} \tag{6.75}$$

Secondly, the expected normalized energy used partial derivative with respect to α is

derived.

$$\frac{\partial}{\partial\alpha}\mathbb{E}[E^N] = (1-\rho)(\lambda r_{Setup} - (\lambda + k\gamma)r_{Idle})\frac{\partial}{\partial\alpha}\frac{\alpha}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}$$

$$\Rightarrow \quad \frac{\partial}{\partial\alpha}\mathbb{E}[E^N] = (1-\rho)(\lambda r_{Setup} - (\lambda + k\gamma)r_{Idle})\frac{\cancel{k\alpha\gamma} + \cancel{\alpha\lambda} + \lambda\gamma - \cancel{k\alpha\gamma} - \cancel{\alpha\lambda}}{(k\alpha\gamma + \alpha\lambda + \lambda\gamma)^2}$$

$$\Rightarrow \quad \frac{\partial}{\partial\alpha}\mathbb{E}[E^N] = (1-\rho)(\lambda r_{Setup} - (\lambda + k\gamma)r_{Idle})\frac{\lambda\gamma}{(k\alpha\gamma + \alpha\lambda + \lambda\gamma)^2} \tag{6.76}$$

Lastly, the expected switching rate partial derivative with respect to α is derived.

$$\frac{\partial}{\partial\alpha}\mathbb{E}[Sw] = (1-\rho)\lambda\gamma\frac{\partial}{\partial\alpha}\frac{\alpha}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}$$

$$\Rightarrow \quad \frac{\partial}{\partial\alpha}\mathbb{E}[Sw] = (1-\rho)\lambda\gamma\frac{\cancel{k\alpha\gamma} + \cancel{\alpha\lambda} + \lambda\gamma - \cancel{k\alpha\gamma} - \cancel{\alpha\lambda}}{(k\alpha\gamma + \alpha\lambda + \lambda\gamma)^2}$$

$$\Rightarrow \quad \frac{\partial}{\partial\alpha}\mathbb{E}[Sw] = (1-\rho)\lambda\gamma\frac{\lambda\gamma}{(k\alpha\gamma + \alpha\lambda + \lambda\gamma)^2} \tag{6.77}$$

Substituting (6.75), (6.76), and (6.77) into (6.74) yields:

$$\frac{\partial}{\partial\alpha}\mathcal{C} = \frac{\lambda\gamma}{(k\alpha\gamma + \alpha\lambda + \lambda\gamma)^2}\Bigg[\frac{k\gamma + \lambda}{\gamma} + \frac{k(k-1)\gamma}{2\lambda} + (1-\rho)[\beta_1(\lambda r_{Setup} - (\lambda + k\gamma)r_{Idle}) + \beta_2\lambda\gamma]\Bigg]. \tag{6.78}$$

Wishing to minimize $\mathcal{C}$ (with respect to α), (6.78) is set to 0.

$$0 = \frac{\lambda\gamma}{(k\alpha\gamma + \alpha\lambda + \lambda\gamma)^2}\left[\frac{k\gamma+\lambda}{\gamma} + \frac{k(k-1)\gamma}{2\lambda} + (1-\rho)[\beta_1(\lambda r_{Setup} - (\lambda + k\gamma)r_{Idle}) + \beta_2\lambda\gamma]\right]$$

$$\Rightarrow \quad 0 = \frac{k\gamma+\lambda}{\gamma} + \frac{k(k-1)\gamma}{2\lambda} + (1-\rho)[\beta_1(\lambda r_{Setup} - (\lambda + k\gamma)r_{Idle}) + \beta_2\lambda\gamma] \quad (6.79)$$

Here a result which was known previously to be true, is seen, but with additional detail. In general, the partial derivative of $\mathcal{C}$ with respect to α cannot equal 0. This implies that $\mathcal{C}$ is minimized when α is at one of its bounds (0 or ∞). This corresponds to the server remaining on, or instantly turning off when it idles. This result was observed through the properties of the Poisson process, but having it present itself in the detailed analysis makes for an interesting observation and sanity check.

While it was previously known that the optimal value of α would lie on its bounds, it was unknown under what conditions each of these bounds would be optimal. It is known that while (6.78) is positive, it is optimal to have $\alpha = 0$. On the other hand, if (6.78) is negative it is optimal to have $\alpha \to \infty$. Which case is optimal can be determined by rearranging the terms of (6.79), to easily see when (6.78) is negative or positive. This rearrangement yields the following inequality.

$$k + \frac{\lambda}{\gamma} + \frac{k(k-1)\gamma}{2\lambda} + (1-\rho)(\beta_1\lambda r_{Setup} + \beta_2\lambda\gamma) \geq \beta_1(1-\rho)(\lambda + k\gamma)r_{Idle} \quad (6.80)$$

When (6.80) holds it is optimal to leave the server on, while if it does not hold, it is

optimal to turn the server off. This gives one the ability to make the optimal decision with respect to α. However, criteria for choosing the second decision variable, k, have not yet been considered.

There are a few ways in which the corresponding optimal value of k can be determined. Firstly, one could set (6.80) to be an equality, which gives a quadratic in terms of k. From here k could be solved, which would give the smallest value of k in which it is optimal to turn the server off (if such a positive real k exists, otherwise it is optimal to keep the server on). The reader is reminded that in practice k must be an integer, and rounding the calculated value may be required. Although this value of k is the smallest value in which it is optimal for the value of α to approach ∞, a larger value of k may exist for which the cost function is lower (k may not be optimally chosen). To determine this optimal value, increasing values of k could be substituted into $\mathcal{C}$ with $\alpha \to \infty$ until the function values increase from one k to the next $(k+1)$. Once this occurs, the value of k is known to be optimal. Secondly, and perhaps the more elegant method, the partial derivative of $\mathcal{C}$ with respect to k can be taken and set to 0.

$$\frac{\partial}{\partial k}\mathcal{C} = \frac{\partial}{\partial k}\mathbb{E}[R] + \beta_1 \frac{\partial}{\partial k}\mathbb{E}[E^N] + \beta_2 \frac{\partial}{\partial k}\mathbb{E}[Sw] \tag{6.81}$$

Again, for the sake of clean algebra, each partial derivative is derived individually. Firstly, the partial derivative of the expected response time with respect to k is determined.

$$\frac{\partial}{\partial k}\mathbb{E}[R] = \frac{\partial}{\partial k}\left(\frac{\alpha}{\gamma}\frac{k\gamma+\lambda}{k\alpha\gamma+\alpha\lambda+\lambda\gamma} + \frac{\alpha\gamma}{2\lambda}\frac{k(k-1)}{k\alpha\gamma+\alpha\lambda+\lambda\gamma}\right)$$

$$\Rightarrow \quad \frac{\partial}{\partial k}\mathbb{E}[R] = \frac{\alpha}{\gamma}\frac{\gamma(k\alpha\gamma + \alpha\lambda + \lambda\gamma) - k\alpha\gamma^2}{(k\alpha\gamma + \alpha\lambda + \lambda\gamma)^2} + \frac{\alpha\gamma}{2\lambda}\frac{(2k-1)(k\alpha\gamma + \alpha\lambda + \lambda\gamma) - k(k-1)\alpha\gamma}{(k\alpha\gamma + \alpha\lambda + \lambda\gamma)^2}$$

$$\Rightarrow \quad \frac{\partial}{\partial k}\mathbb{E}[R] = \alpha\lambda\frac{\alpha + \gamma}{(k\alpha\gamma + \alpha\lambda + \lambda\gamma)^2} + \frac{\alpha\gamma}{2\lambda}\frac{k^2\alpha\gamma + (2k-1)(\alpha\lambda + \lambda\gamma)}{(k\alpha\gamma + \alpha\lambda + \lambda\gamma)^2} \tag{6.82}$$

Secondly, the partial derivative of the expected normalized energy used with respect to k is derived.

$$\frac{\partial}{\partial k}\mathbb{E}[E^N] = (1-\rho)\alpha\frac{\partial}{\partial k}\frac{\lambda r_{Setup} - (\lambda + k\gamma)r_{Idle}}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}$$

$$\Rightarrow \frac{\partial}{\partial k}\mathbb{E}[E^N] = (1-\rho)\alpha\frac{-\gamma r_{Idle}(k\alpha\gamma + \alpha\lambda + \lambda\gamma) - (\lambda r_{Setup} - (\lambda + k\gamma)r_{Idle})\alpha\gamma}{(k\alpha\gamma + \alpha\lambda + \lambda\gamma)^2}$$

$$\Rightarrow \frac{\partial}{\partial k}\mathbb{E}[E^N] = (1-\rho)\alpha\lambda\gamma\frac{\alpha(r_{Idle} - r_{Setup}) - (\alpha + \gamma)r_{Idle}}{(k\alpha\gamma + \alpha\lambda + \lambda\gamma)^2}$$

$$\Rightarrow \frac{\partial}{\partial k}\mathbb{E}[E^N] = -(1-\rho)\alpha\lambda\gamma\frac{\alpha r_{Setup} + \gamma r_{Idle}}{(k\alpha\gamma + \alpha\lambda + \lambda\gamma)^2} \tag{6.83}$$

Lastly, the partial derivative of the expected switching rate with respect to k is derived.

$$\frac{\partial}{\partial k}\mathbb{E}[Sw] = (1-\rho)\alpha\lambda\gamma\frac{\partial}{\partial k}\frac{1}{k\alpha\gamma + \alpha\lambda + \lambda\gamma}$$

$$\Rightarrow \frac{\partial}{\partial k}\mathbb{E}[Sw] = -(1-\rho)\alpha\lambda\gamma\frac{\alpha\gamma}{(k\alpha\gamma+\alpha\lambda+\lambda\gamma)^2} \tag{6.84}$$

Before any further work is done, as a sanity check each partial derivative is inspected to ensure the mathematics agree with the known optimal values presented in Table 5.6. It is known that $\mathbb{E}[R]$ is minimized when k is at its lower bound. When (6.82) is examined, one observes that there is a decision variable (k in this case), present for the first time in the numerator of a partial derivative. However, due to the restriction that $k \geq 1$, this expression is always positive, and can never equal 0. This of course implies that the optimal value of k lies on its lower bound, which agrees with previous observations. For the partial derivatives of $\mathbb{E}[E]$ and $\mathbb{E}[Sw]$, it was previously known that these would be minimized as k approaches infinity. This is the exact result which is observed in (6.83) and (6.84), as both expressions are always negative, implying that the optimal value of k lies at its upper bound. Therefore, all sanity checks pass, and the work proceeds to derive the optimal values for the cost function $\mathcal{C}$.

Substituting (6.82), (6.83), and (6.84) into (6.81) yields:

$$\begin{aligned}\frac{\partial}{\partial k}\mathcal{C} = \frac{\alpha\lambda\gamma}{(k\alpha\gamma+\alpha\lambda+\lambda\gamma)^2}\Bigg[&\frac{\alpha}{\gamma}+1+\frac{1}{2\lambda^2}(k^2\alpha\gamma+(2k-1)(\alpha\lambda+\lambda\gamma)) \\ &-(1-\rho)(\beta_1(\alpha r_{Setup}+\gamma r_{Idle})+\beta_2\alpha\gamma)\Bigg].\end{aligned} \tag{6.85}$$

Setting the previous equation equal to 0 allows one to solve for the optimal value of

k.

$$0 = \frac{\alpha\lambda\gamma}{(k\alpha\gamma + \alpha\lambda + \lambda\gamma)^2}\left[\frac{\alpha}{\gamma} + 1 + \frac{1}{2\lambda^2}(k^2\alpha\gamma + (2k-1)(\alpha\lambda + \lambda\gamma)) - (1-\rho)(\beta_1(\alpha r_{Setup} + \gamma r_{Idle}) + \beta_2\alpha\gamma)\right]$$

$$\Rightarrow \quad 0 = \frac{\alpha}{\gamma} + 1 + \frac{1}{2\lambda^2}(k^2\alpha\gamma + (2k-1)(\alpha\lambda + \lambda\gamma)) - (1-\rho)(\beta_1(\alpha r_{Setup} + \gamma r_{Idle}) + \beta_2\alpha\gamma)$$

$$\Rightarrow \quad 0 = k^2\left(\frac{\alpha\gamma}{2\lambda^2}\right) + k\left(\frac{\alpha+\gamma}{\lambda}\right) + \left[\frac{\alpha}{\lambda} + 1 - \frac{\alpha+\gamma}{2\lambda} - (1-\rho)(\beta_1(\alpha r_{Setup} + \gamma r_{Idle}) + \beta_2\alpha\gamma)\right] \tag{6.86}$$

Equation (6.86) is quadratic in k, and can be used to solve for optimal values of k for general values of α. However, as previously shown in this section, α will either be 0 or ∞ in the optimal policy. Furthermore, when $\alpha = 0$, the choice of k is trivial. Therefore, the optimal value of k can be determined under the assumption that $\alpha \to \infty$. While one can take the limit as $\alpha \to \infty$ of (6.85), it is simpler to start from the partial derivatives, and retake them under the assumption that $\alpha \to \infty$.

Again, the partial derivative of $\mathbb{E}[R]$, with the assumption of $\alpha \to \infty$ is the first one solved.

$$\frac{\partial}{\partial k}\mathbb{E}[R] = \frac{\gamma}{2\lambda}\frac{\partial}{\partial k}\left(\frac{k(k-1)}{k\gamma + \lambda}\right)$$

$$\Rightarrow \quad \frac{\partial}{\partial k}\mathbb{E}[R] = \frac{\gamma}{2\lambda}\left(\frac{(2k-1)(k\gamma+\lambda) - k(k-1)\gamma}{(k\gamma+\lambda)^2}\right)$$

$$\Rightarrow \quad \frac{\partial}{\partial k}\mathbb{E}[R] = \frac{\gamma}{2\lambda}\left(\frac{k^2\gamma + (2k-1)\lambda}{(k\gamma+\lambda)^2}\right) \tag{6.87}$$

Secondly, the partial derivative of $\mathbb{E}[E^N]$, with the assumption of $\alpha \to \infty$ is solved.

$$\frac{\partial}{\partial k}\mathbb{E}[E^N] = (1-\rho)\frac{\partial}{\partial k}\frac{\lambda r_{Setup} - (\lambda + k\gamma)r_{Idle}}{k\gamma+\lambda}$$

$$\Rightarrow \quad \frac{\partial}{\partial k}\mathbb{E}[E^N] = (1-\rho)\frac{-\gamma r_{Idle}(k\gamma+\lambda) - (\lambda r_{Setup} - (\lambda+k\gamma)r_{Idle})\gamma}{(k\gamma+\lambda)^2}$$

$$\Rightarrow \quad \frac{\partial}{\partial k}\mathbb{E}[E^N] = -(1-\rho)\frac{\lambda\gamma r_{Setup}}{(k\gamma+\lambda)^2} \tag{6.88}$$

Lastly, the partial derivative of $\mathbb{E}[Sw]$, with the assumption of $\alpha \to \infty$ is solved.

$$\frac{\partial}{\partial k}\mathbb{E}[Sw] = (1-\rho)\lambda\gamma\frac{\partial}{\partial k}\frac{1}{k\gamma+\lambda}$$

$$\Rightarrow \quad \frac{\partial}{\partial k}\mathbb{E}[Sw] = -(1-\rho)\lambda\gamma\frac{\gamma}{(k\gamma+\lambda)^2} \tag{6.89}$$

This time, substituting (6.87), (6.88), and (6.89) into (6.81) yields:

$$\frac{\partial}{\partial k}\mathcal{C} = \frac{\lambda\gamma}{(k\gamma+\lambda)^2}\left[\frac{k^2\gamma + (2k-1)\lambda}{2\lambda^2} - (1-\rho)(\beta_1 r_{Setup} + \beta_2\gamma)\right]$$

Setting the previous equation equal to 0 allows one to solve for the optimal value of

k while $\alpha \to \infty$ (the server instantly shuts off when it idles).

$$0 = \frac{\lambda\gamma}{(k\gamma+\lambda)^2}\left[\frac{k^2\gamma+(2k-1)\lambda}{2\lambda^2} - (1-\rho)(\beta_1 r_{Setup}+\beta_2\gamma)\right]$$

$$\Rightarrow \quad 0 = \frac{k^2\gamma+(2k-1)\lambda}{2\lambda^2} - (1-\rho)(\beta_1 r_{Setup}+\beta_2\gamma)$$

$$\Rightarrow \quad 0 = k^2\frac{\gamma}{2\lambda^2} + k\frac{1}{\lambda} - \left[\frac{1}{2\lambda} + (1-\rho)(\beta_1 r_{Setup}+\beta_2\gamma)\right] \tag{6.90}$$

From here, one has a quadratic in k which can be solved to find the value which minimizes $\mathcal{C}$ as $\alpha \to \infty$. One should note that the ceiling and floor of the value of k, which solves the quadratic, must be taken and substituted into $\mathcal{C}$ to see which value is lower. One should also note that although this gives the optimal value of k, this does not guarantee that $\alpha \to \infty$ is a configuration of the optimal policy, and the value of $\mathcal{C}$ with $\alpha = 0$ must also be checked against.

Although there are several cases to consider, the equations and observations presented in this section give one the ability to derive the optimal policy for the weighted sum cost function. Hopefully the methods and ideas presented here make it clear to the reader how other cost functions can be minimized using the expressions derived in this work.

6.2 Constrained Optimization

All considerations up to this point assumed that one wished to minimize a cost function with no imposed constraints. However, in practice this may not be the case. For example, one may have some constraints on metrics such as the expected response time. This is seen commonly in service level agreements (SLAs) between parties, where one party guarantees its customer a certain mean response time. So, a natural problem that arises in this context is to minimize energy costs while satisfying the SLA. When this scenario is formulated into a linear optimization problem, it is found that the optimal value of α does not necessarily lie on one of its bounds, unlike the previous analysis and applications. As this is the primary point being made here, the rest of the problem is left as simple as possible, and is set in the domain of an $M/M/1 \circ \{M, M, 1\}$ queue.

Consider the problem of minimizing $\mathbb{E}[E^N]$ for an $M/M/1 \circ \{M, M, 1\}$ queue while having $\mathbb{E}[R]$ be less than or equal to some threshold, denoted by T. The problem is assumed to be feasible and non-trivial, that is $\frac{1}{\mu-\lambda} < T < \frac{1}{\mu-\lambda} + \frac{1}{\gamma}$. In other words, the threshold is not less than the server's expected response time when it always remains on, but on the other hand, the solution also is not one which immediately turns the server off. Furthermore, to fully ensure the problem is non-trivial, the assumption that $r_{idle} < \frac{\lambda}{\lambda+\gamma} r_{setup}$ must also be imposed. This is due to the result stated in Theorem 3, which implies that if this condition does not hold, then $\mathbb{E}[E^N]$ is minimized when $\mathbb{E}[R]$ is minimized, which is when the server always remains on.

This linear optimization problem is seen formally as:

$$\begin{aligned} min. \quad & \mathbb{E}[E^N_{M/M/1}] + (1-\rho)\frac{\alpha}{\alpha\gamma + \alpha\lambda + \lambda\gamma}(\lambda r_{Setup} - (\lambda+\gamma)r_{Idle}) \\ s.t. \quad & T \geq \mathbb{E}[R_{M/M/1}] + \frac{1}{\gamma}\left(\frac{\alpha\gamma + \alpha\lambda}{\alpha\gamma + \alpha\lambda + \lambda\gamma}\right) \\ & \alpha \geq 0, \end{aligned} \tag{6.91}$$

where the single decision variable is α. This formulation can be easily solved directly without employing any well known algorithms such as simplex, Newton's method, trust region, etc. The observation is made that the objective function decreases in α (based on the assumptions on r_{Setup} and r_{Idle}). Furthermore, the expected response time (the right hand side of the inequality in the first constraint), increases in α. It follows from this that the objective function is minimized when (6.91) is an equality. Specifically, the objective function is minimized when,

$$\begin{aligned} & T = \mathbb{E}[R_{M/M/1}] + \frac{1}{\gamma}\left(\frac{\alpha(\lambda+\gamma)}{\alpha\gamma + \alpha\lambda + \lambda\gamma}\right) \\ \Rightarrow \quad & \gamma(T - \mathbb{E}[R_{M/M/1}])(\alpha\gamma + \alpha\lambda + \lambda\gamma) = \alpha(\lambda+\gamma) \\ \Rightarrow \quad & \alpha(\lambda+\gamma)(1 - \gamma(T - \mathbb{E}[R_{M/M/1}])) = \lambda\gamma^2(T - \mathbb{E}[R_{M/M/1}]) \\ \Rightarrow \quad & \alpha = \frac{\lambda\gamma^2}{\lambda+\gamma}\left(\frac{T - \mathbb{E}[R_{M/M/1}]}{1 - \gamma(T - \mathbb{E}[R_{M/M/1}])}\right). \end{aligned} \tag{6.92}$$

Here it can be seen that α can take on a large range of non-trivial values. Specifically, in the setting of constrained optimization it can be optimal to leave the server idling for some amount of time. In contrast to the non-constrained context, these results are quite different. The optimal value of α not being at one of its bounds has many

implications, and raises more than a few questions. Firstly, the shape of the idling time distribution now has a greater effect on the overall system behaviour in the optimal policy. Secondly, the question of whether to keep track of idling times between busy periods now becomes non-trivial. Lastly, out of all possible shapes the idling time distribution can have, deriving which one is optimal could be a daunting task. Although these observations and concerns are interesting, they are out of the scope of this work, and are left for future research.

6.3 Sleep States

The presented model has until now assumed that the sever has exactly two energy states that it can be set to (*on* and *off*). However, modern servers usually have several discrete sleep settings which they can be set to. While in these sleep states, the server consumes a lower amount of energy than being idle but, like being turned off, it cannot process jobs. The advantage of switching the server to one of these sleep states, instead of turning it completely off, is that typically if the server is "sleeping" rather than being off, it takes less time to turn on. This section considers servers which have sleep states, and extends the model to see what kind of policies can be derived.

A class of policies, $\mathcal{P}$, is defined, which exhibit very similar behaviour to the policies which have been considered previously. Policies in class $\mathcal{P}$ wait for k jobs to accumulate in the queue while in a lower energy state before beginning to turn on. Once turned on, the system processes jobs until it becomes idle. If the system idles for a certain amount of time before a new job arrives, it moves to the same lower energy

state that it started in, and repeats its behaviour. The key difference here is that now there exists different lower energy states (the sleep states), and the server is restricted to only use one of them. It will be shown that the model can be used to find the optimal policy contained in $\mathcal{P}$.

The following extensions are made to the previous model.

- The server has I different sleep states it can be set to, where the ith sleep state is denoted by $SLEEP_i$. As stated before, jobs cannot be processed while the server is in state $SLEEP_i$, $\forall i : 0 < i \leq I$. For each state $SLEEP_i$, there is a corresponding energy cost, denoted $E_{Sleep,i}$ (along with an energy ratio with respect to E_{Busy}, $r_{Sleep,i}$). For each sleep state there also exists a corresponding turn on rate, denoted γ_i. Typically, $\forall i : 0 < i < I . E_{Sleep,i} \leq E_{Sleep,i+1}$ and $\gamma_i \leq \gamma_{i+1}$. In other words, if a sleep state uses more energy than another, then it is also expected to take less time to turn on than the one which uses less energy.

- Instead of moving to the energy state OFF after a given idling time, it instead transitions to some energy state $SLEEP_i$. Here the steady state probabilities of $\pi^i_{0,0}$ to $\pi^i_{0,k-1}$ now correspond to the steady state probabilities of being in state $SLEEP_i$ rather than OFF. Furthermore, when k jobs arrive to the system and the system enters the energy state $SETUP$, it transitions to the energy state $BUSY$ with rate γ_i rather than γ.

To analyse this system, two variations must also be made to the expressions derived in Chapter 5. Firstly, all instances of γ in the equations for $\mathbb{E}[R]$, $\mathbb{E}[E^N]$, and $\mathbb{E}[Sw]$ (for whatever distributional assumptions have been made) must be changed to γ_i.

Secondly, a slight addition must be made to the expression for $\mathbb{E}[E^N]$, (5.53), to account for the energy now being consumed in the sleep state.

$$\mathbb{E}[E^N_{Sleep,i}] = \mathbb{E}[E^N] + (1-\rho)\frac{k\alpha\gamma_i}{k\alpha\gamma_i + \alpha\lambda + \lambda\gamma_i} r_{Sleep,i}$$

From here, one can analyse the system and obtain the optimal values of α and k for a system where the lower energy state it moves to is $SLEEP_i$. Substituting these values into the cost function gives the minimum value for the cost function, denoted opt_i, under the assumption that the lower energy state used by the system is $SLEEP_i$. Once one has all I of these corresponding optimal values, by iterating through the sleep states, one can simply take the minimum of them, as well as opt_{Off} (the minimum of the cost function if OFF is used as the lower energy state). With this minimum opt value, a policy can be designed to always transition to the corresponding energy state of $SLEEP_i$, or OFF. This policy is the optimal policy in $\mathcal{P}$.

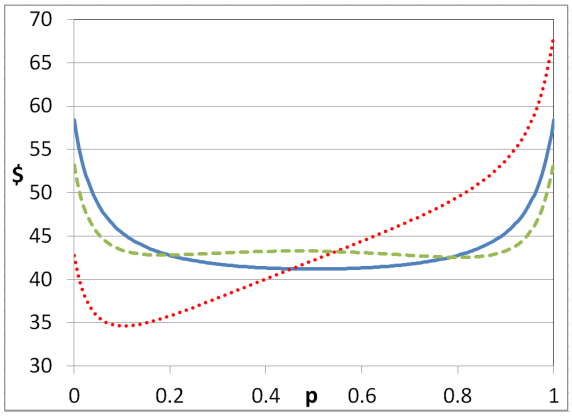
Although accounting for the sleep states of the server allows one to derive improved policies than if they were to be ignored, it can no longer be claimed that this model can describe the true optimal policy. In other words, the optimal policy may not be in $\mathcal{P}$. This is due to the fact that the optimal policy may have the server be in some sleep state until k_1 jobs accumulate, then move to a higher sleep state where it waits for k_2 jobs to accumulate before turning on. However, when the optimal values of k are low for any individual sleep state under the analysis, it is conjectured that the policy will be close to, if not optimal.

6.4 Random Routing

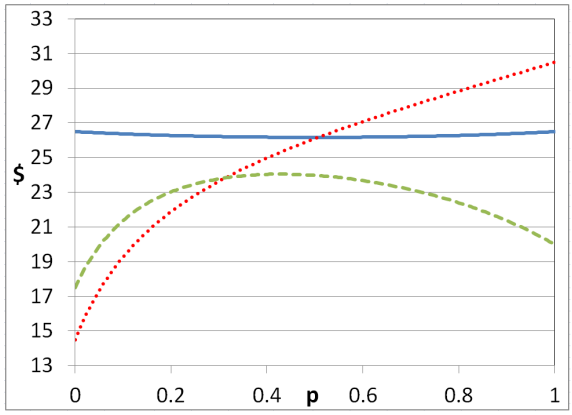
Here the model is applied to a multi-server setting where random routing is employed. As will be seen, although the model assumes a single server, the random routing context still allows for analysis. Consider a system with two $M/M/1 \circ \{M, M, k\}$ queues. When a job arrives to the system, it is sent to the first queue with probability p and is sent to the second queue with probability $(1-p)$. When optimizing against some cost function, there now exists five decision variables, α_1, α_2, k_1, k_2, and p, where the subscripts 1 and 2 denote the values for the first and second server, respectively. It is known that the values for α_1 and α_2 will be either set to 0 or approach ∞, which breaks the problem into three cases (due to symmetry) where we instead look to optimize against k_1, k_2 and p and then take the lowest value from among the three cases. The cases are classified as follows. The first is $\alpha_1 = \alpha_2 = 0$, the second is $\alpha_1 \to \infty$ and $\alpha_2 = 0$, and the third is $\alpha_1 \to \infty$ and $\alpha_2 \to \infty$.

We wish to minimize $\mathbb{E}[N] + \beta\mathbb{E}[E]$. This falls within the class of cost functions, (4.5), as $\mathbb{E}[N]$ can be scaled to give $\mathbb{E}[R]$ and here it is in fact scaled by a unit constant of dollars/jobs. It is well known that for the first case since the servers will always be on and each server will be in $BUSY$ for $\frac{p\lambda}{\mu}$ and $\frac{(1-p)\lambda}{\mu}$ proportion of time respectively, that the optimal configuration in that case is to set $p = 0.5$, i.e. balance the loads. It will be seen that the other cases provide non-trivial and interesting optimal values for p.

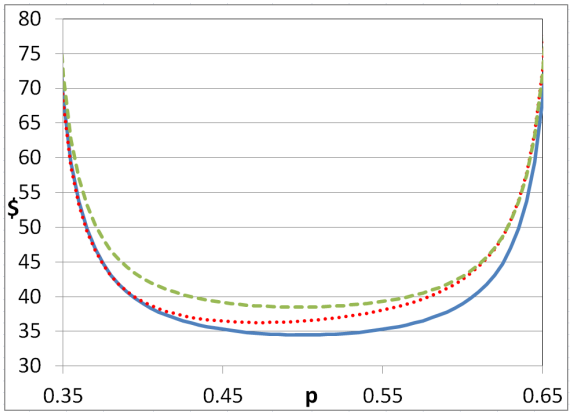
Figure 6.13 shows several examples under different parameter configurations of the



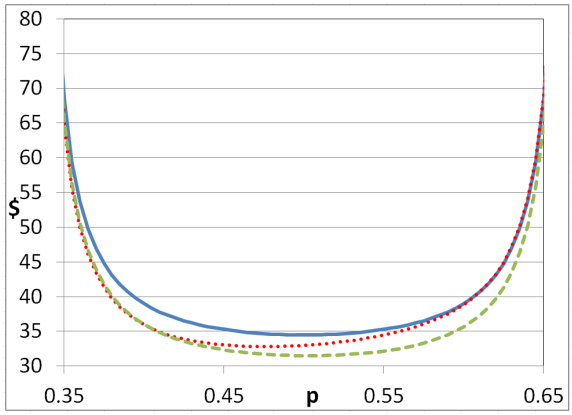
(a) $\mu = 2, \lambda = 1.9, \gamma = 0.2, \beta = 22, k_1 = 5, k_2 = 7$



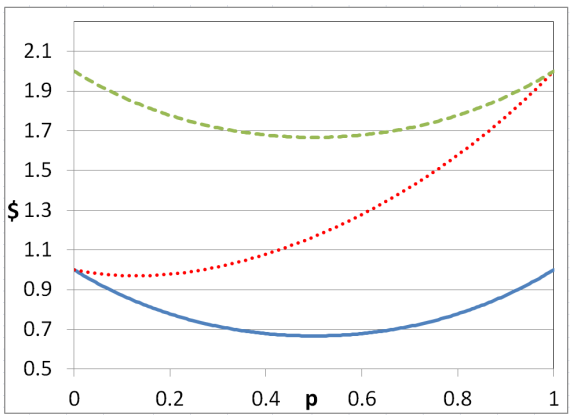
(b) $\mu = 2, \lambda = 1, \gamma = 0.2, \beta = 15, k_1 = 1, k_2 = 4$



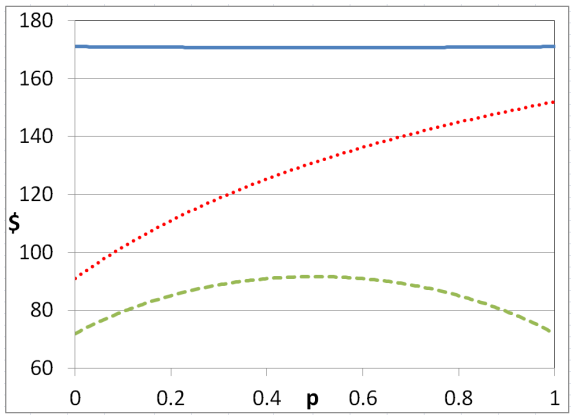
(c) $\mu = 2, \lambda = 3, \gamma = 0.5, \beta = 15, k_1 = 3, k_2 = 2$



(d) $\mu = 2, \lambda = 1.9, \gamma = 3, \beta = 15, k_1 = 1, k_2 = 2$



(e) $\mu = 2, \lambda = 1, \gamma = 1, \beta = 0, k_1 = 1, k_2 = 1$



(f) $\mu = 2, \lambda = 1, \gamma = 1, \beta = 100, k_1 = 1, k_2 = 1$

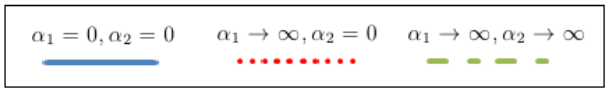


Figure 6.13: Random Routing – Optimization vs p

cost function versus p in the three different cases where the optimal k values are used, and r_{idle} and r_{setup} are both set to 0.8 (this implies it would never be optimal from an energy stand point to leave the servers on). Figure 6.13-(a) shows a medium loaded system where either server could take all of the arrivals and still be stable. Here it is seen that the optimal server configuration is to have a server which is always on which takes the majority of the system load (89.5%), while a server which turns off when it becomes idle takes a small portion of the system load (10.5%). This means that a lot of the time, the server that turns off will just remain off with up to four jobs waiting in the queue. This may seem unfair to the jobs which are "unlucky" enough to be put into this queue but this is an unfortunate side effect of energy concerns in this setting.

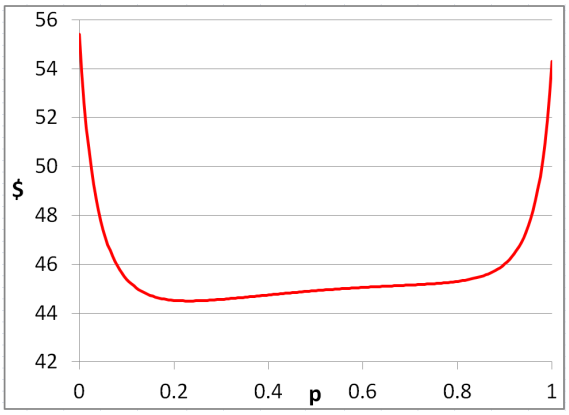
Figure 6.13-(b) shows a lightly loaded system. Perhaps unsurprisingly, the optimal configuration is still such that one server remains on and one turns off. However, the server which turns on and off is completely ignored. In other words, the configuration which optimizes the random routing problem is simply an M/M/1 queue. This is somewhat expected since the load on the system is so light it is not advantageous to use the second server. This result is interesting when put in the context of having many servers to choose from. This result would imply that under certain parameter configurations, a (potentially) large set of servers would be ignored, or remain off. This implies that for certain configurations, adding servers to the system has no bearing on the optimal policy.

Figure 6.13-(c) and Figure 6.13-(d) show the results for a heavily loaded system where both servers must be used or the system will be unstable. The curves of the three

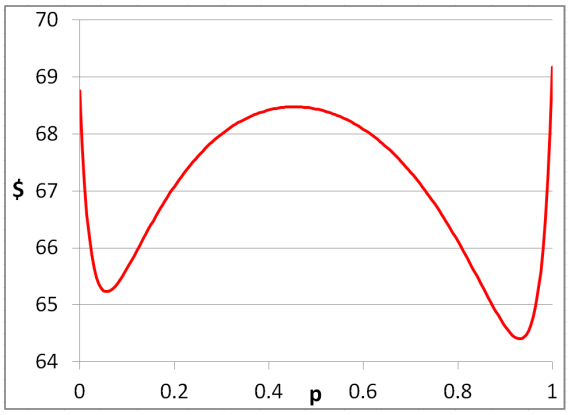
cases here begin to converge to similar shapes. In Figure 6.13-(c), where the setup rate is relatively low ($\gamma = 0.5$), the classical load balancing approach gives the best configuration with both servers always on and $p = 0.5$. It is observed that as the setup rate of the server increases ($\gamma = 3$), both servers being on becomes sub-optimal and the case of both servers turning on and off begins to dominate. In fact, the optimal value is $p = 0.505$ and not $p = 0.5$ as one might expect. Having the servers shut down as γ increases is quite intuitive, since the faster the server can turn on, the more appealing it is to shut it off.

As was seen in Figure 6.13, simple load balancing is not sufficient to arrive at the optimal configuration for multi-server systems using random routing, since non-trivial values of p that optimize the system were shown to exist. Taking a more narrow look at the single case of having both servers able to turn off in Figure 6.14, shows a similar non-trivial result. Here the graphs also become asymmetric with respect to p, and furthermore the optimal values of k_1 and k_2 are not equal. As in the case of having one server always on, and one server able to turn off, load balancing is not optimal. It is noted that if load balancing were used in Figure 6.14-(b), i.e. $p = 0.5$, the result would be a disaster, as it is one of the worst configurations possible in this context. Adding energy concerns to these systems greatly impacts the complexity of the analysis as typical load balancing algorithms (which are used in practice) are no longer optimal.

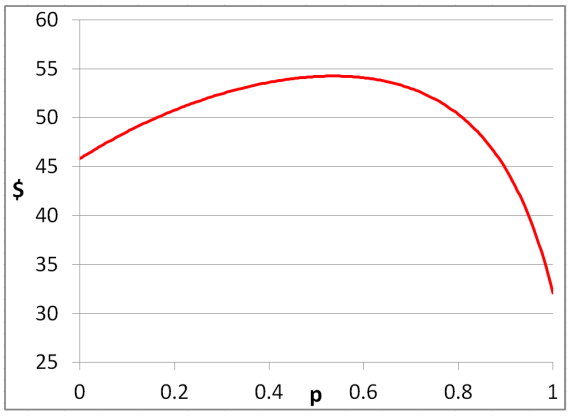
This also raises questions on the implications for other multi-server settings such as round robin routing or in an $M/M/c \circ \{M, M, k\}$ queue. Specifically, there is no



(a) $\mu = 2, \lambda = 1.9, \gamma = 0.1, \beta = 15, k_1 = 6, k_2 = 3$



(b) $\mu = 2, \lambda = 1.9, \gamma = 0.1, \beta = 30, k_1 = 3, k_2 = 4$



(c) $\mu = 2, \lambda = 0.5, \gamma = 0.1, \beta = 50, k_1 = 7, k_2 = 1$

Figure 6.14: Random Routing – Single Case

reason why in general each server should be identical with respect to the server's α and k values. This would make finding the optimal policy for such systems a much harder problem than others may have previously thought, as the number of decision variables grows with the number of servers in the system. However, although this implies the multi-server system has a much greater complexity than the single server system, by no means does this imply that these problems are intractable.

Chapter 7

Conclusions

As energy costs of servers as well as the relative energy consumed by servers increase, industry must put a greater emphasis on determining optimal policies. Here an exact analysis was given of the single server systems $M/M/1 \circ \{M, M, k\}$ and $M/G/1 \circ \{G, M, k\}$, with respect to $\mathbb{E}[N]$, $\mathbb{E}[R]$, $\mathbb{E}[E]$, and $\mathbb{E}[Sw]$ as well as analysis for an $M/G/1 \circ \{G, G, k\}$ queue with respect to $\mathbb{E}[E]$ and $\mathbb{E}[Sw]$. This gave us an array of tools, equations, and results to arrive at optimal policies for many single server energy-aware systems under general settings. This analysis was also leveraged in several other applications, such as SLA optimization, servers with sleep states, and a multi-server system with random routing. For the latter it was shown that typical load balancing algorithms are not enough to arrive at an optimal configuration. Furthermore, this context gives a deeper insight into the analysis of these energy-aware multi-server systems with other routing policies. In particular, heterogeneous servers may be desirable, in contrast to models where energy costs are not considered. Energy factors will always be present in these systems and it is important that we gain as much insight and understanding into these problems as possible.

7.1 Future Work

While this work presented many useful and broad contributions, there still remains much work to be done. The field is rich with many open problems which have immediate applications. Even considering the work done here, there are many natural extensions which are of interest. Firstly, further work can be done on the analysis of single server systems. Several variations were discussed in Chapter 6, where for some the model fell short of deriving the optimal policy. Some of these variations are as follows.

- Alter the model to account for different discrete energy states which the server can be set to. This includes sleep states, as well as speed scaling. Sleep states were already discussed in Chapter 6. Speed scaling refers to server having the option of being set to "higher" energy states where the processing rate, μ, is increased.

- Do further analysis on constrained optimization. As mentioned earlier, when the cost function is constrained, the behaviour of these optimal parameters and policies changes drastically. Many interesting questions of how these constraints impact the system remain to be answered, such as what type of turn-off criteria define the optimal policies, and how are these criteria influenced by the cost function?

- Relax some or all of the model assumptions. The model makes two assumptions about the system. Firstly, the model assumes the jobs are processed by a first come first serve (FIFO) policy. Secondly, it is assumed that shutting the server down happens instantly. It would be of interest how relaxing one or both of

these assumptions would affect the optimal policies.

- Allow for the derivation under all cost functions. While this work was able to derive the optimal policy for a broad range of cost functions, some remain unknown, eg. $\mathbb{E}[R \cdot E]$.

Secondly, and perhaps the more daunting extension, is to analyse the system under a general number of servers. When deriving the optimal policy, allowing for a general amount, say c servers, greatly increases the complexity of the analysis. As discussed in the context of random routing, the number of decision variables will increase as the number of servers increases. Furthermore, when dealing with something like an $M/M/c \circ \{M, M, k\}$ queue, even more decision variables must be introduced. Specifically, a parameter for the threshold number of jobs in the queue before a server begins to turn off instead of processing a new job must be introduced, as this value does not equal 0, as it did in an $M/M/1 \circ \{M, M, k\}$ queue.

Bibliography

[1] J. R. Artalejo. A unified cost function for M/G/1 queueing systems with removable server. *Trabajos de Investigacion Operativa*, 7(1):95–104, 1992.

[2] L. A. Barroso and U. Holzle. The case for energy-proportional computing. *Computer*, 40(12):33–37, Decemeber 2007.

[3] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam. Managing server energy and operational costs in hosting centers. *SIGMETRICS Performance Evaluation Review*, 33(1):303–314, June 2005.

[4] U.S. EPA. Report to congress on server and data center energy efficiency. Technical report, U.S Environmental Protection Agency, 2007.

[5] S. W. Fuhrmann and R. B. Cooper. Stochastic decompositions in the M/G/1 queue with generalized vacations. *Operations Research*, 33:1117–1129, 1985.

[6] A. Gandhi, S. Doroudi, M. Harchol-Balter, and A. Scheller-Wolf. Exact analysis of the M/M/k/setup class of markov chains via recursive renewal reward. In *ACM SIGMETRICS*, 2013.

[7] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. A. Kozuch. Optimality analysis of energy-performance trade-off for server farm management. *Performance Evaluation*, 67(11):1155–1171, November 2010.

[8] A. Gandhi and M. Harchol-Balter. M/M/k with exponential setup. Technical report, Carnegie Mellon University, 2010.

[9] A. Gandhi, M. Harchol-Balter, and I. Adan. Server farms with setup costs. *Performance Evaluation*, 67(11):1123–1138, November 2010.

[10] D. Gross and C. M. Harris. *Fundamentals of Queueing Theory*. Wiley-Interscience, third edition, 1998.

[11] M. Harchol-Balter. *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. Cambridge University Press, 2013.

[12] L. Kleinrock. *Queueing Systems*, volume One. Wiley-Interscience, 1975.

[13] K. Li. Optimal power allocation among multiple heterogeneous servers in a data center. *Sustainable Computing: Informatics and Systems*, 2(1):13–22, 2012.

[14] J. D. C. Little. A proof for the queuing formula: L = λW. *Operations Research*, 9(3):383–387, 1961.

[15] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. H. Andrew. Greening geographical load balancing. In *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, SIGMETRICS, pages 233–244, 2011.

[16] Michele Mazzucco and Dmytro Dyachuk. Optimizing cloud providers revenues via energy efficient server allocation. *Sustainable Computing: Informatics and Systems*, 2(1):1–12, 2012.

[17] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*. Dover Publications Inc., 1994.

[18] A. Penttinen, E. Hyytia, and S. Aalto. Energy-aware dispatching in parallel queues with on-off energy consumption. In *IEEE International Performance Computing and Communications Conference*, pages 1–8, November 2011.

[19] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs. Cutting the Electric Bill for Internet-Scale Systems. In *ACM SIGCOMM*, Barcelona, Spain, August 2009.

[20] J. Slegers, N. Thomas, and I. Mitrani. Dynamic server allocation for power and performance. In *Proceedings of the SPEC international workshop on Performance Evaluation: Metrics, Models and Benchmarks*, SIPEW '08, pages 247–261, Berlin, Heidelberg, 2008. Springer-Verlag.

[21] N. Tian and Z. G. Zhang. *Vacation Queueing Models Theory and Applications*. Springer Science, 2006.

[22] A. Wierman, L. L. H. Andrew, and M. Lin. *Handbook on Energy-Aware and Green Computing*, chapter Speed Scaling: An Algorithmic Perspective, pages 385–406. CRC Press, 2012.

[23] A. Wierman, L. L. H. Andrew, and A. Tang. Power-aware speed scaling in processor sharing systems. In *Proceedings of INFOCOM*, 2009.

[24] X. Xu and N. Tian. The M/M/c queue with (e, d) setup time. *Journal of Systems Science and Complexity*, pages 446–455, 2008.