Predicting Waiting Times in Telephone Service Systems

Research Thesis
Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Operations Research and Systems Analysis

Efrat Nakibly

Submitted to the Senate of the Technion – Israel Institute of Technology
Av 5762 Haifa July 2002

The research was carried out under the supervision of Prof. Avishai Mandelbaum in the Faculty of Industrial Engineering & Management.

I sincerely thank Prof. Mandelbaum for guiding me through the course of the research, for his patience and for his willingness to assist at any time.

Special thanks to Prof. Isaac Meilijson from the School of Mathematics, Faculty of Exact Sciences in Tel-Aviv University for allowing me to continue his work, for helpful discussions and for his time.

I thank also Prof. Haya Kaspi for the time and efforts invested in reviewing this work and for her helpful comments.

The generous financial help of the Technion Graduate School is gratefully acknowledged.

Contents

List of Figures

List of Tables

Abstract

Customer satisfaction is today the major concern for almost all companies. Businesses compete not only on quality of products but on service level as well. Service level is determined by various attributes. Customer surveys in service systems demonstrate that the waiting time is one of the key factors when evaluating service level.

Waiting might cause feelings of frustration, anger, boredom and uncertainty. There is a clear tradeoff between waiting times and operational costs of the service system. Shortening waiting times usually requires the increasing of staffing levels. Surveys and research suggest that customer satisfaction with waiting can be improved by managing customer expectations of the waiting and customer perception of the waiting, even without shortening the waiting time itself.

Providing information about the anticipated waiting is an example of an activity that can significantly improve customer satisfaction with the waiting. Information may be provided upon arrival, and updated periodically thereafter. Providing information about anticipated delays may also result in objective improvements of the performance of the service system. With this information, customers can decide whether they are willing to wait or to leave and come back later. Abandonment after waiting is replaced with balking upon arrival. The number of customers in the system decreases and so does the percentage of customers who find a congested or a full system.

Information regarding the anticipated waiting is of a special importance in service systems with invisible queues. In such systems, the uncertainty involved in waiting is higher than in visible queues, and it does not decrease over time. Customers have no means to estimate queue lengths or progress rate, and the feelings of frustration and anxiety increase during the waiting.

The most common examples of service systems with invisible queues are telephone call centers. Call centers are used to provide a wide range of services such as information, customer support, marketing and more. Call centers have become today's main service channel, and are estimated to handle about 70% of all customers interactions.

The goal of our work is to propose methods for estimating waiting times in service systems in general, and in call centers in particular. We model the call center as a queueing system. We use exact analytical methods, approximations and simulations in order to estimate the waiting times for different schemes of call centers. Since the goal is to provide information which is relevant to a specific customer at a specific time, we focus on estimating the waiting time given the system state at the time of estimation. This is different than estimating the overall performance of the system, such as the average waiting time of all customers, which is usually done assuming a steady-state.

First, we estimate waiting times for classic queueing models that maintain a simple First-Come-First-Served service discipline. Then, we focus on systems with priority service disciplines and with skills-based routing. In those systems, calls are assigned to agents based on pre-defined rules, considering required service type and servers skills. The problem of estimating waiting times in such systems is particulary challenging, since the number of calls that will enter service before the call of interest depends on future processes (arrivals and service) and is not known.

We analyze in details a relatively simple case of two servers and two service types, assuming exponential times (service, and inter-arrivals), and assuming out abandonment and retrials. We suggest two methods for an exact estimation of the waiting times: difference equations and matrix geometric solutions. We demonstrate the computational complexity in each of the methods, and explain under which conditions they are applicable.

Motivated by the complexity of exact calculations, we suggest an inexplicit analytical approximation. The approximation may be implemented for a wide range of models, assuming Poisson arrivals and exponential service times. It can be implemented where servers are not statistically identical, as well as where there are different types of customers. The approximation is iterative, with each iteration representing the time between two successive completions of service. The changes in the system during each iteration (arrivals, abandonment) are taken into account, while replacing accurate distributions with deterministic values or with simple distributions.

We use formal analysis, as well as simulations, in order to evaluate the approximation accuracy. Numerical comparisons to simulation results indicate that the approximation works very well in many cases. However, we also identify the weaknesses of the approximation, and explain when it is expected to yield good results, and when improvements could still be made.

List of notations and abbreviations

$\overset{d}{=}$ - equal in distribution

$A \perp B$ - A and B are independent

1_x - indicator of the event x. The variable receives the value 1 if the event x occurs, and the value 0 otherwise.

$M_{g_x}(\theta)$ - generating moment function of g_x with parameter θ

$L_{g_x}(\theta)$ - Laplace transform of g_x with parameter θ

SD - standard deviation

$f_X(x)$ Density function of variable X in point x.

FCFS - first come first served

w.p. - with probability

Chapter 1

Introduction

1.1 Background

Over the last decades, customers and customer satisfaction have become the major concern of almost all companies. Investments in customer retention have been steadily increasing, and businesses now compete not only on quality of products but on service level as well. Businesses are attempting to establish on-going and long-term relationships with customers by providing various services (prior to transaction, during transaction and after transaction).

The service level is determined by qualitative attributes, such as professionalism of the serving agent, as well as quantitative, such as the process duration. Some of the attributes, such as the service time or whether the service was completed in the first interaction, are objective, and some of them, such as agent's politeness, or one's feeling that the service was too long, are subjective. Queueing models are often used to calculate operational attributes of the service level: service times, waiting times, number of people in the system, percentage of abandoning customers and more.

1.1.1 The role of waiting times

Very often, the service process involves delays. The time waiting for service is acknowledged as one of the most critical attributes of service level. Customer surveys in service systems demonstrate that waiting time is a key factor when evaluating quality of service ([16, 29, 13, 30, 14, 8, 6]). Long waits often result in feelings of anger and in a low customer satisfaction. Waiting time is also correlated with other quantitative measures, such as queue size and abandonment.

Obviously, there is a tradeoff between the number of agents that provide service and the operational service level, for example, waiting times. Shortening waiting times, thus, directly increases operational costs. Waiting time is, in fact, one of the main considerations when determining staffing levels ([6, 28, 15] and more). A common method is to plan for the least number of agents that suffice to satisfy a required service level. Such planning can be based on analytical modelling or on simulations.

In addition to the waiting duration itself, customer satisfaction is also affected by the perceived waiting time and by the waiting experience [17, 14, 16, 29, 30]. Waiting for some services takes place while the customer is on-line waiting (a face-to-face service, or a telephone service) or when customers continue their regular activities (waiting for an e-mail reply). Different factors contribute to the waiting experience: waiting conditions; the interest level while waiting (filled time vs. empty time); the feeling of justice (or of injustice) in the service discipline and more. Surveys demonstrate that customer satisfaction can be improved without changing the waiting time itself, but by managing customer expectations or by improving the waiting experience. Some examples of activities that may improve the waiting experience are: playing music while waiting; providing information or other services while waiting; making sure the physical waiting environment is comfortable (in face-to-face service); explaining the reasons for waiting; and providing information regarding the anticipated waiting time.

1.1.2 Informing customers about anticipated delays

Information about anticipated waiting time reduces the uncertainty and increases customer satisfaction [13, 14]. It may also shorten the perceived waiting time. Different types of waiting information can be provided to customers: queue-size, waiting time of the longest-waiting customers or the anticipated waiting time of an individual customer. Surveys have indicated that different types of waiting information should be provided under different circumstances [13].

Information about anticipated waiting times also objectively improves the service level [31]. Customers decide, right upon arrival if they are willing to wait. As less customers decide to abandon after already waiting for a while, the steady-state number of customers in queue decreases and so does the percentage of customers who find the system full.

Since we are dealing with stochastic systems, there is no possible way to predict the exact waiting time. The best one can do is estimate the waiting time *distribution*. The service system manager should then decide what is the exact information that will be provided to customers. For example, he may decide to provide the mean of the estimated waiting time or any other percentile of the distribution. Informing on a short waiting time, which is likely to underestimate the actual waiting, might reduce the reliability of the service provider in the eyes of the customers. Informing on a long waiting time, on the other hand, might result in longer perceived waiting times and in decrease in satisfaction. Another issue is the cost of waiting. This cost consists both of the physical costs of waiting (such as call time in telephone systems) and on decrease in customer satisfaction. The decision of what quantile of the waiting time distribution to inform, depends on the desired outcome. If the manager wishes to decrease the load in the system, to minimize operational costs or to control traffic, she may decide to inform of a high percentile and a longer waiting time. If, however, the purpose is to keep customers in the system, she may choose to inform of a shorter waiting time (associated with a lower percentile). Doing that, the manager risks in building non-realistic expectations of the waiting time. These expectations affect both customers' satisfaction with the waiting [17] and their patience [35]. Choosing to inform of a "too-short" waiting time might result in a decrease in customers' satisfaction and in loss of trust.

Waiting information has a particulary important role in service systems with invisible queues. When queues are visible (for example, in a face-to-face service), customers typically see upon arrival both how many people are waiting and how many agents provide services. After waiting for a while, they may learn the service rates. The level of uncertainty decreases and they may even be able to estimate the remaining waiting time. When the queue is invisible, the uncertainty involved in waiting is much higher, and does not decrease over time. In [4], Cleveland and Mayben describe the difference in the waiting experience between visible and invisible queues. They suggest that when the queue is visible, customers experience dissatisfaction upon arrival, as they see that there is a queue. Then, as they are advancing in queue, in a satisfactory rate, the feelings of dissatisfaction decrease until they receive service and happily leave the system. Where as in queues that are invisible, customers do not experience dissatisfaction upon arrival, but as they are kept on hold, feelings of anger and dissatisfaction emerge. These feelings intensify until they eventually possibly abandon. Providing waiting information in these cases may eliminate the gap between reality and customer expectations.

1.1.3 Call centers

Service systems with invisible queues are very common. The most common examples are telephone call centers, which have enjoyed a growing popularity over the last decade. As will be explained later, in this work we focus on estimating waiting times in telephone call centers.

Most generally, a call center is a service system in which agents (servers) serve customers, remotely over the phone. Call centers are used to provide services in many areas and industries: emergency centers, information centers, help-desks, tele-marketing, and more. A telephone service enables customers to obtain a fast response, with a minimal effort. Providing services via call centers, instead of a face-to-face service, usually translates into lower operational costs to the service provider.

The call center industry has been steadily growing. Estimates (GeoTel (1998), Vantive (1996)) indicate that around 70% of all customer transactions occur in call centers (see www.callcenternews.com/resources/statistics). IDC (1999) estimated that the worldwide call center market generated $23 billion in revenues in 1998, and was projected to double to $58.6 billion by 2003 (www.callcenternews.com/ s2industryfacts). The growth in the call-center industry has been observed worldwide: According to a Tele-Management Search (TMS), in 2000 there were around 7 million call center agents in the US, with an expected annual growth rate of 20%. The number of call centers was estimated to be 69,500 and to reach 78,000 in 2003 (see www.incoming.com). Call Center Week reports a 24% annual growth rate for call centers in Canada, and estimates the number of Canadian call centers to be 6,500 (www.incoming.com). In UK [7], call centers employed 1.7% of the working population, or nearly 400,000 people in 1999 . Over the years 1994-1999 in UK, customer calls to large organizations have roughly doubled.

Terminology of call centers is somewhat different from that of a general service system: lines are used in the telephone world to indicate the number of places in the system (service and waiting), busy signal for the case when a customer calls, finds a full system and leaves, and a call is a single arrival to the system.

Contemporary call centers provide a wide range of services to a wide range of customers. Individual agents can often provide only several types of the services. Computerized system assist them with customer information and history. Using routing systems, agents often know who is the customer and what type of service is required, before even picking up the phone. Some of the call centers (for example tele-marketing) initiate outgoing calls and not only answer incoming calls. Call centers are now in the process of evolving towards contact centers, where services are remotely provided not only via phone but also via fax, e-mail and on-line chatting (internet).

Due to the special characteristics of telephone call centers, they often operate under very heavy loads. In a large call center, which is well managed and planned [15], many hundreds of agents can answer thousands of calls a day. Agents utilization could reach $90\% - 95\%$, and still customers hardly experience busy signals, and about 50% of the customers are answered immediately.

Customers are usually willing to tolerate much shorter waits for a telephone service than for face-to-face. Abandonment rates might be much higher and customers who abandon may re-try shortly after. In such an environment, information about anticipated waiting times may significantly change the performance of the call center. Queue lengths decrease and so does the percentage of busy-signals [31]. In contemporary call-centers, the direct waiting costs are often paid by the company (1-800 numbers). Therefore, balking right after being informed on the anticipated time may also dramatically decrease the operational costs.

We believe that, once an accurate method for estimating waiting times is available, it will be fairly easy to implement in call centers a system informing the customer of that estimated time, if desirable. We know of call centers in which customers are informed, upon calling, about the number of customers waiting ahead of them and of the elapsed waiting-time of the longest-waiting-customer [21]. That implies that the technology for the suggested application exists and is being only partially used. Customers are offered only objective measured data, without performing any estimation.

1.2 Goals

In this work, we aim to develop methods for estimating waiting times in telephone-call-centers, operated under different schemes. As we have already explained, the best one can do is to estimate the waiting time *distribution* and leave the decision of what percentile to inform of to the call-center's manager. However, as will be seen later in our work, we will not always be able to provide the full distribution of the estimated waiting time. For some models, we only estimate the mean, or the mean and the variance, of the anticipated waiting time. We focus on call centers that handle incoming telephone calls only, but the methods and the results may be applied to other service systems as well.

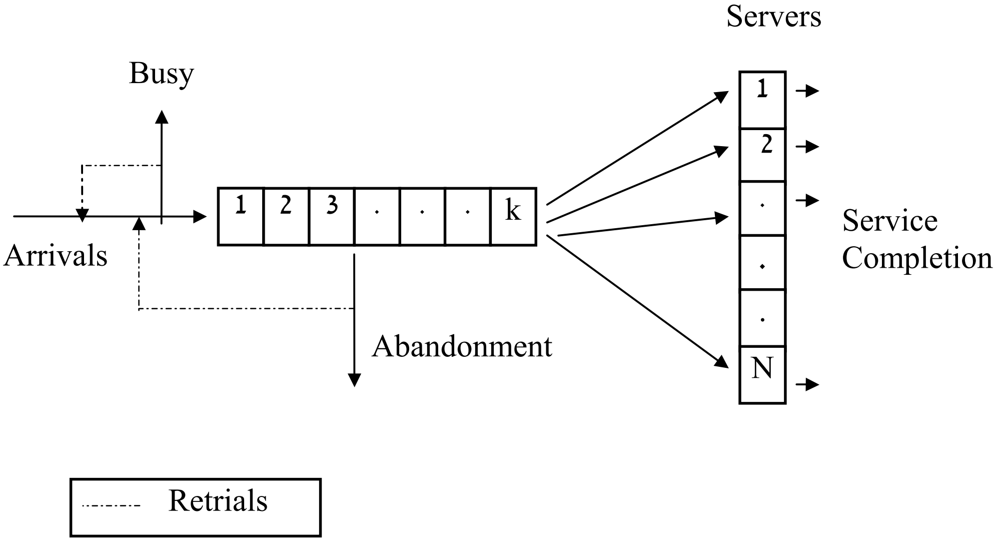
1.3 Method

In order to analyze a call-center, we model it as a queueing system. An arriving call is immediately answered (enters service) if there is an available agent, joins the queue if all agents are busy and not all lines are busy, and is blocked (receiving a busy signal) if all lines are busy. Figure 1.1 illustrates this description.

Obviously, this description is often too simple to fit a real-life call center, where many agents provide services to many customers. Some examples are agents with different skills that can not provide all the services, or service that consist of more than a single phase. Yet, the simple scheme in Figure 1.1 provides a natural starting framework. The analysis of call centers as queueing system has been widely in use. It is possible to analyze the system both by analytical models or by simulations. A review of the main existing models can be found in [15].

We focus on estimating the waiting times, given the system state at the time of estimation. This is different than estimating general performance of the system, such as the average waiting time of all customers, which is usually done assuming a steady-state. We demonstrate the use of different estimation methods for both exact calculations and approximations.

Figure 1.1: A simple call center as a queueing system



First, we estimate waiting times for classic queueing models, that maintain a simple First-Come-First-Served service discipline. Then, we focus on systems with priority service discipline and with skills-based routing. We use difference equations and matrix geometric solutions to analyze in details a relatively simple case. Motivated by the complexity of exact calculations, we suggest an inexplicit analytical approximation. Last but not least, we use simulations and compare their results to those of our approximations.

1.4 Contents

The rest of this work is organized as follows: In Chapter 2 we briefly summarize the literature relevant to waiting times in service systems, as well as some papers regarding analytical models of call-centers. In Chapter 3 we present the principles and some examples of estimating waiting times in queueing systems. The rest of the work focuses on systems with priorities. In Chapter 4 we use difference equations to estimate the waiting time in a relatively simple model, with only two identical servers and two service types. Then, in Chapter 5 we demonstrate how the matrix geometric method can be used to estimate waiting time in systems with priorities. We explain when and how the method can be applied. We deduce that explicit estimations might be too complicated for a more general case and in Chapter 6 we suggest an inexplicit analytical approximation. The suggested approximation method was first developed by Prof. Isaac Meilijson, from the

School of Mathematics in the Faculty of Exact Sciences, Tel-Aviv University, and is further elaborated on here. In Chapter 7 we present mathematical justifications for parts of the approximation, (for the simple case analyzed previously). The detailed algorithm and some extensions to it are presented in Chapter 8. In addition, we evaluate the accuracy of our approximation in different systems by numerically comparing its results with those of a simulation. The results and conclusions are presented in Chapter 9, as well as some additional analysis of the waiting time behavior, based on the simulation results.

Chapter 2

Literature Survey

In this chapter we provide a brief overview of the literature regarding waiting times in general and call centers in particular. We will mention papers that discuss three aspects of the subject:

i. The role of waiting times (delays) in service systems.

ii. Informing customers about anticipated delays.

iii. Analytical models of call centers.

To find more about call centers, we refer the readers to [18], which is an extensive list of over 200 references to papers dealing with different aspects of call centers.

2.1 The role of waiting times in service systems

Waiting time is one of the most common quantitative performance measures of a service system, having a direct effect on both customer satisfaction and operational costs. The discussion of waiting times involves various aspects from different arenas including Psychology, Marketing, and Operations Research. To name a few: the relation between waiting time and customer satisfaction; The effect of the waiting environment on the perceived waiting time and on customer satisfaction; The relation between waiting time and other operational measures such as abandonment; and staffing decisions which consider the tradeoff between staffing levels and waiting times. These are only a few examples of issues that are of interest when studying the role of waiting times. We now mention some of the works that deal with the above subjects.

- Maister in [17] discusses the psychology of waiting lines. The satisfaction of a customer is determined by the difference between his expectation and his perception. In order that customers will be more satisfied with the service, the service provider should manage customers' perceptions and customers' expectations. Maister suggests 8 propositions, that can be used by service organizations to influence their customers' satisfaction with waiting times:

  - Unoccupied time feels longer than occupied time.
  - Pre-process waits feel longer than in-process waits.
  - Anxiety makes waits seem longer.
  - **Uncertain waits are longer than known finite waits.**
  - Unexplained waits are longer than explained waits.
  - Unfair waits are longer than equitable waits.
  - The more valuable the service, the longer one will wait.
  - Solo waiting feels longer than group waiting.

- Larson [16] studies the queueing experience, in order to identify factors that affect the waiting experience and the user's perception, in addition to the delay time itself. He suggests that the waiting experience is not a linear function of the waiting itself, and that it is affected by physiological and psychological factors. One of the points raised is that information regarding the anticipated delay, increases user satisfaction and decreases the frustration in waits. Larson speaks both about the importance of initial information and the importance of seeing that queue is moving. Some additional guidelines are suggested: customers are expecting social justice in queues, that usually means a FCFS discipline, and are getting irritated by 'skips and slips'; A pleasant and interesting waiting environment contribute to customer satisfaction; Unfilled time pass slower than busy time.

  The paper focuses on service provided face-to-face and not on tele-queues.

- In Taylor [29], the relationship between delays and evaluation of service is considered, motivated by the belief that service waits can be controlled by either operations management or by perceptions management. Results indicate that longer delays lead to lower evaluation of service. The delay creates feelings of both anger and uncertainty. These feelings are affected by additional factors as well as by the delay duration itself. For example, anger is affected by the control level that service provider has on the delay. Decreasing the anger level, may moderate the effect of the delay on customer satisfaction. The research is conducted by examining the relations between flight delays and customer satisfaction. It is, therefore, mostly relevant to delays of services that have been scheduled for a certain time.

- Additional study of the effect of waiting conditions in telephone systems on the perceived waiting time and on customer satisfaction is presented in Tom, Burns and Zeng [30]. The findings indicate that waiting conditions affect customer satisfaction. More precisely, filled time (such as listening to music while waiting) is better than empty time. Both customer satisfaction and the perception of the service provider as customer-oriented may be increased by giving the callers a selection of what they want to listen to, or by pre-selecting items that fit customer taste. Surprisingly, though waiting conditions had a clear effect on customer satisfaction, they did not always affect the perceived waiting time, especially in cases where people may call at any convenient time. That might be explained by the fact that while waiting, people continue other non-related activities.

- In [8], Feinberg describes a study aimed to identify factors that affect customer satisfaction in call centers. The study utilizes data of 514 call-centers in US. The results indicate that the queue-time indeed affect customer satisfaction. Yet, the variables with the greatest contribution to customer satisfaction (out of 13 variables measured) are percentage of calls closed on first contact and abandonment rate.

- The cost of waiting time is referred to by Carmon, Shanthikumar and Carmon in [3]. They illustrate how aspects of psychological cost of waiting can be incorporated into an analytical queueing model. The analysis of such a model can lead to changes in the design of service schemes. More specifically, they assume that customers' dissatisfaction from waiting can be reduced by providing parts of the service in an early stage. They use an analytical model to show that service should sometimes be divided when it can be provided in multiple separate phases.

- Davis in [6] refers to the tradeoff between waiting times and costs in service systems. The total cost of having a customer wait is derived from two separate cost components: the cost of providing the service and the cost associated with the customer wait. Cost associated with customer wait results from dissatisfaction of a customer and may lead to fewer future visits at the system (of the customer or of his friends). Davis suggests to define a waiting cost function and then to determine the optimal staffing level together with the optimal waiting time, minimizing the total cost of the system. This is an alternative to the more common approach of determining the minimal staffing level required to satisfy a certain waiting time (referred to as a constraint).

- Borst, Mandelbaum and Reiman [1] also discuss the staffing problem as an optimization problem, considering both waiting cost and agents' cost. We refer to this paper again with more details, as well as to additional works regarding methods of defining the required staffing level, in Section 2.3.

2.2 Informing customers about anticipated delays

Providing information about anticipated delays affects both customer satisfaction and customer behavior. First, it decreases the level of uncertainty, but it also offers customers means to decide whether or not they are willing to wait. This decision quantitatively affects the number of customers in the system, system load and waiting times.

- The psychological effects of waiting information are discussed in Hui and Tse [13]. An experimental study was conducted to examine the impact of waiting information on service evaluation. The authors distinguish between two types of waiting information: waiting duration (how long one is expected to wait) and queue information (how many people are in line ahead of him). The impact of providing waiting information (of each type) on service evaluation is examined for cases of short-wait, intermediate-wait and long-wait conditions. Hui and Tse define three mediators between waiting information and service evaluation: perceived waiting time, effective response to the wait and acceptability of the wait. The results indicate that the effect of waiting information on service evaluation is mediated mainly by effective response and by acceptability, and not by the perceived waiting time. The wait-information that should be provided to customers varies by the wait duration. In short waits, no information is needed. When the wait is intermediate, waiting duration information appears to be a better choice than queueing information. When the wait is long, queueing information may be better than wait duration information. The reason for this is that when informing customers of long wait duration, they might get dissatisfied with the time lost. The experiment was conducted when waiting for a process to be completed by a computer (and not at a call center). Wait information updates were provided continuously.

- Katz and Larson [14] describe an empirical study conducted in a bank (face-to-face service) to examine customer perception of waiting in line and examine methods for making waiting more tolerable. Motivated by the understanding that when it comes to customers satisfaction perception is the reality, the authors investigate methods of perception management. They suggest to change customer perceptions and customer expectations rather than the actual performance of the bank. Two methods were examined: an electronic news-board was installed in order to make the waiting more interesting, and a clock informing customers about the anticipated delay, before they enter the queue. The study resulted in some interesting findings: first, customers tend to overestimate the time they spend in queue. As actual waiting time increased overall customer satisfaction tended to decrease, stress level tended to increase and so did perceived waiting time and the "reasonable" time customers are willing to wait. Longer perceived waiting times were associated with lower satisfaction levels.

As to methods of changing customer perception: news-board had a small effect on perceived waiting times, but it made time spent in line more plausible. Informing by a clock about the anticipated delay resulted in shorter perceived waiting times. The authors suggest two possible explanations for that. Customers may have believed what the clock told them about waiting times and adjusted their perceptions. Alternatively, the clock may have made customers more aware of time. However, it was not found that the clock improved customers overall satisfaction.

Another finding is the relation between information about anticipated delays and balking rate. With the information about the anticipated delays, more people looked into the bank, saw the clock, and left.

- Carmon and Kahenman study in [2] the effect of queue length information and of queue speed on the waiting experience. They refer to cases in which information regarding one's position in queue is constantly available. It is found that at the beginning of waiting the level of (dis)satisfaction is determined by the queue length, but as time go on the speed in which the queue is moving becomes the dominant factor. During waiting, positive responses are observed with each movement of the queue. These responses deteriorate between movements. The retrospective evaluation of the waiting experience is mainly affected by the feelings at the end of the waiting rather than by the feelings when joining the queue. In summary, the initial queue length affects expectations. The speed of queue relatively to those expectations, and especially the speed experienced near the end of waiting, have the major contribution the overall evaluation.

- Mandelbaum, Sakov and Zeltyn [21] study empirical data of the call center of an Israeli bank. An activity of an entire year, including more than 440,000 calls, was analyzed. The call center provides several service types, and applies a priority policy, upon which high priority customers are advanced by a 1.5 minutes in the queue (if there is a queue). Various aspects of the call center, such as the arrival process, queueing times, abandonment and service times were studied. The analysis also refers to the time customers are willing to wait. The findings are interesting for understanding the implications of informing customers about anticipated delays. Upon arrival to queue and about every minute or so thereafter, customers were exposed to an automatic message informing them of their place in queue (relatively to the number of servers) and of the waiting time of the longest-waiting customer. From the hazard rate function for time customers were willing to wait, it was found that the messages caused customers to abandon (with peaks after the first two messages, and lower picks later on). A possible explanation is that the message "reminds" customers of their waiting and causes an acceleration of the subjective time flow.

- In his article [31], Whitt analyzes qualitative implications of informing users about anticipated delays on the call center performance. The call center is modelled by a Markovian birth and death (BD) process. It is assumed that each customer is willing to wait a fixed amount of time before beginning service, called the delay threshold. Delay thresholds of successive customers are exponential with a mean of α^{-1}.

  Whitt compares two models: Model 1, with no wait information provided. With a probability β customers are not willing to wait at all and therefore balk (if there is no server available). Otherwise they join the queue but might renege after exponential time with a mean of α^{-1} (provided that service has not been started by that time). Model 2 refers to the case in which wait-information is provided. Being informed of the anticipated delay, customers can choose if they are willing to wait or leave the system immediately. Reneging is therefore entirely replaced by balking. Based on steady-state analysis, Whitt compares the number of customers in the system for each of the two models. The number in system in Model 1 is shown to be larger than in Model 2, in the likelihood-ratio stochastic ordering. That implies that blocking is higher in Model 1, and the probability to be served without waiting is higher in Model 2. Numerical comparisons show that the performance of the two systems, measured by percentage of eventually served customers and by the waiting time for served customers, are remarkably similar. The difference, however, is that with balking instead of reneging customers who do not receive service do not waste time waiting.

- Whitt [33] deals with the problem of estimating the anticipated waiting time of individuals, given the system state at the time of estimation. The subject is therefore very similar to that of our work, and the paper is very relevant. Whitt presents both accurate methods and approximations of waiting times.

Different models are analyzed such as: different service types, abandonment and non-exponential waiting times. All the models discussed in that paper assume first-come-first-served (FCFS) service discipline. The methods and the results for some of the cases are detailed in Chapter 3 of this work .

2.3 Analytical models of call centers

Naturally, call centers are often viewed as queueing systems. We shall discuss two general schemes of models:

i. First-come-first-served (FCFS) service

ii. Skills-based routing

2.3.1 Models with FCFS discipline

In a detailed introduction to call centers by Koole and Mandelbaum [15], it is explained how call centers can be modelled by queueing systems of various characteristics. Various results and models with references are mentioned in that paper. The authors examine models of single type customers and single skill agents; models with busy signals and abandonment; models with multiple intervals and overloads; skills-based routing; call blending and multi-media; and geographically dispersed call centers. For each of the above, the authors describe and explain the common models and provide references to detailed studies. The paper provides an extensive summary of the main existing models of call centers.

We now explain in more details some of the main models. We rely mainly on [15], and are assisted by several additional relevant works.

General schemes of call center modelling

- The simplest model is M/M/s, also known as Erlang C, which assumes exponential service times and Poisson arrivals. Usually, Erlang C is too simple to describe a real-life call center (for example, it assumes out abandonment and busy signals).

- Non-exponential service times leads to the M/G/s queue. The system can be approximated by the M/M/s model and the variance of service times. For example [28],

$$E[wait\ for\ M/G/s] \quad \approx \quad E[wait\ for\ M/G/s] \cdot \frac{1+C^2}{2}, \qquad (2.1)$$

where C is the coefficient-of-variation of the service time, denoted by $C = E/\sigma$.

- Heavy-traffic approximations are often helpful. For example, in the M/G/s queue with a small to medium number of servers, s, the waiting time is approximately exponential. The case of a large s gives rise to different asymptotic behavior, and is discussed in Halfin and Whitt [12] for G/M/s queue and in Puhalskii and Reiman [25] for M/PH/s (service times with a phase-type distribution).

At the extreme, call centers can be managed and operated according to a quality driven regime (almost all customers are served immediately upon calling, but agents utilization might be low) or according to an efficiency driven regime (agents are utilized almost 100% of their time, but customers might usually experience long delays) (Borst, Mandelbaum and Reiman [1]). However,well managed call centers can operate within a rationalized regime, where quality and efficiency are balanced. In a large call center operated under a rationalized regime, agents utilization could reach 95%, when about half of the customers are served immediately upon calling.

- Theory that supports rationalized regime was first developed by Halfin and Whitt [12]. In that paper the relation between number of agents, agent utilization and probability of delay, in heavy traffic systems with a large number of servers ($s \uparrow \infty$) is formulated.

- Borst, Mandelbaum and Reiman in [1], develop a framework for asymptotic optimization of queueing system. They consider both agents cost and waiting cost (or quality of service) and determine the optimal number of servers N^*. They then formally introduce and explain the square-root staffing principle, which has been used long before. In the simplest form, if c is the hourly cost of an agent, and α is the hourly cost of customers' delay, then $N^* = R + y(\frac{a}{c})\sqrt{R}$, when R is the offered load, and y is some function discussed in the paper.

Abandonment and busy signal

An important tradeoff in call centers is that between busy signals, long waits and abandonment occurring due to long waits. For a given number of servers and a given load, that tradeoff is determined by the number of lines. A large number of waiting places results in long waits and in abandonment. A small number of waiting places results in a high percentage of calls that get a busy signal, and do not succeed in entering the call center queue.

- Garnett, Mandelbaum and Reiman [10] study the subject of abandonment. The simplest abandonment model, M/M/s/B+M, is analyzed, and an asymptotic analysis of the M/M/s+M model in the Halfin-Whitt regime is suggested. One of the main results is the relation between the number of agents, the offered load, probability of delay and probability of abandonment in heavy traffic systems, when $s \uparrow \infty$. The result obtained extend the findings of Halfin and Whitt [12], accommodating abandonment. Then, some rules of thumb for staffing level and for estimating performance measures under a quality-driven, efficiency-driven and rationalized regimes are suggested.

- In Zohar, Mandelbaum and Shimkin [35] the authors investigate the relation between customer patience and waiting times, and conclude that customer patience depends on the mean waiting time in the queue, and on customer experience regarding that waiting time. In particular, they suggest that the exponential assumption often used for abandonment is unjustified.

Non-Poisson or non-stationary arrival process

- Standard modelling of call centers use measures in steady-state for each time-interval, typically 30 minutes or 1 hour. Koole and Mandelbaum explain that though this works, in general, pretty well, exceptions arise when overload occurs in one or more intervals. They refer the readers to [20], [19] and to additional works that study such overloads.

- Sometimes, the structure of call centers (for example, a network queueing scheme or the appearance of re-trials) leads to a non-Poisson arrival process. In Woodside, Stanford and Pagurek [34] a predictor of queue lengths and delays of such an G/M/m queue is suggested. N_n is defined as the queue length immediately before the nth arrival, given N_0. N_k is a Markov chain, and it is used to calculate the moments of N_n and of the waiting time of the n'th customer W_n.

2.3.2 Skills-based-routing

Skills-based-routing refers to the on-line strategy that matches callers to agents. Such models are essentials when many servers with different skills provide different service types.

A common way of implementing skills-based routing is by specifying two selection rules [9] [15]: agent selection - how does an arriving call select an idle agent, if there is one; and call selection - how does an idle agent select a waiting call, if there is one. Agents are first divided into groups in a way that all agents within a group have all the skills associated with it (a certain skill may be associated with more than one group). For each skill, there is an ordered list of groups containing that skill. Each type of calls requires that the agent serving it will have certain skills. An arriving call of a certain type is then assigned to the first group (among the groups with the required skills), that has an agent available, or that becomes available. If an available agent can handle each one of several waiting calls, then some priority rule is employed in order to determine which call to handle. It is also possible that a call is assigned to a group only if there is at least a certain number of agents available for service.

We further refer to skills-based routing in Subsection 3.3.1.

The following are works that refer to models of agents with different skills and customers with different characteristics.

- Garnett and Mandelbaum [9] discuss the design of skills-based routing schemes. They explain the main decisions that should be made when implementing a skills-based routing: defining customer types; defining servers skills and numbers; and defining the control policy. They classify some canonical designs of skills-based routing (I, N, X, W, M and V designs). They analyze the performance of each of the canonical designs. The analysis is mainly simulation-based.

- In [27], Schwartz deals with the Lane Selection model. In this model there are n distinct types of customers and n types of service facilities. Each type of service facility can serve some of the service types (known to the customers). Queues to the different facilities are separated (unlike the general scheme of skills-based routing, in which there is one queue and selection is done only before starting service). Upon arrival, customer choose which of the queues to join. Customers' decisions regarding which line to join are assumed to be 'deterministic'. In other words, customer's decision given queue size is known. The model assumes Poisson arrivals and exponential service times (different service types have different means of service times). Based on arrival rates and on lane selection rules, the rate of joining each queue is determined. Then each line is referred to as a classical single-server queue. The author finds the average waiting time for customers of each type for several special cases of the described model.

- Roque [26] refers to [27], and shows that the analysis in [27] is incorrect, since the arrival process to each of the stations (services facilities) is not a Poisson process. Customer decisions depend on the queue size in the different stations. Therefore, arrivals to the different stations are dependent, and, moreover, they are not Poisson.

- A telephone service system with two service types, and three groups of operators (servers) is studied by Perry and Nillson in [24]. The operators are divided into 3 groups: operators who can serve only the first type of customers, operators who can serve only the second type of customers and operators who can serve both types. Namely, this is the **M** model in the language of Garnett and Mandelbaum [9]. The model is different from the one discussed in [27] and [26] by the fact that assignment of calls to servers is done when service starts and not necessarily upon arrival. If there are calls waiting, when a server becomes available, then he is assigned the longest waiting-call (out of the calls he is capable of serving). If servers are free when a call arrives, then it is assigned to the longest-waiting server. The goal of the work is to determine the expected waiting time of each type of customers and the average occupancy level for the three different types of operators. The authors suggest an approximation for the expected waiting time and numerically compare it with simulation results. The method used is partitioning the arriving process into separate arrival streams, accounting for the type of operator that provides the service, and then decompose the heterogeneous service into two M/G/m systems. The approximation seems to overestimate delays and to be more accurate for low-load systems.

- A model of two service types, where some servers can handle both service types (G), and some can handle one type only (R) is discussed again by Green in [11]. (This is referred to as the **N** model, in Garnett and Mandelbaum [9]). The author aims to find the stationary probabilities and to estimate mean delay times. The approach taken is different than those in [24], [27] and [26]. The queue is described as two separate queues: a restricted queue, with all the customers waiting according to their arrival order, and a general queue, that consists of type G customers, who were skipped in line by a type R customer because a type G server was not available. Once such an epoch occur, customers are moved from the restricted queue to the general one. First the model is described by a bi-variate Markov process with states (i, j), $i \geq 0, j \geq 0$, when i is the number of type R customers in service plus the number of customers (either type) in the restricted queue, and j is the number of type G customers in service plus the number of customers in the general queue. The generator matrix of the process has a repetitive geometric form, and the author uses the matrix geometric method in order to find the stationary probabilities.(For more details about matrix geometric method see Neuts [23] and Chapter 5 in this work). The state space of the model described is infinite in both dimensions, but the method of matrix geometric solution can only be applied to systems in which one element as most can get infinite number of values. Therefore, the system is approximated by assuming that there exists an integer k such that the number of type G customers do not exceed k. (When there are already k type G customers in the system, all arrivals are assumed to be of type R). The matrix geometric method is applied to the approximated model and results for stationary distribution are obtained.

- In [32], Whitt discusses the case of several service types, which significantly differ by the required service times. He suggests that an efficient service model for this case is partitioning customer types into disjoint subsets, that will be served separately in multi-server queues. That is, of course, possible only when the required service type is identified upon arrival. The author studies the tradeoff between the economies of scale gained from larger systems and the cost of having customers with short service times wait longer due to customers with much longer service times. It is required, therefore, to formulate an optimization problem, seeking to minimize the total number of servers used, while requiring that each class of service meets a specified performance requirement.

- In Cobham [5], a method for estimating the average waiting time in a system with priorities is suggested for two special cases: a single server, and some service time distribution; and multi-server queues, when service times of all types are exponential with a mean of μ^{-1}.

- In [28], Sze analyzes a quiet general model with large server team sizes, different service time distribution (exponential, hyper-exponential and Erlang), non-stationary Poisson arrivals, abandonment and reattempts and certain priority structure. The author suggests an approximation to determine the required staffing to satisfy an acceptable service level in half-hour intervals. The service level is determined by the delay function of customers of each type.

Chapter 3

Methods and Examples of Estimating Waiting Times

In this chapter we present some examples of estimating the anticipated waiting times in queueing systems with different characteristics. First we discuss simple systems, and then we introduce skills-based routing and systems with static priorities. More specifically, we shall discuss the following subjects:

i. Principles of estimating waiting times

ii. Examples of estimating waiting times: we first present methods of estimating delays for models with a service discipline of first-come-first-served (FCFS).

iii. Non FCFS service disciplines: we discuss priorities and skills-based routing and present methods to estimate delays for models with such service disciplines.

3.1 Principles of estimating waiting times

Generally speaking, there are two ways to estimate waiting times:

i. Based on the system state at a given moment

ii. Based on system state distribution (steady state)

For any of these methods, system states should first be defined. The calculations involved in the first method, are usually easier, but operational effort is high. Since estimations of that type are usually needed on-line, system state should be tracked in real-time. Estimations of the second type are used to predict the general behavior of the system (as opposed to the experience of a specific customer).

The calculations are based on the individual results obtained by the first method averaged with respect to the steady state distribution, when Poisson arrivals are assumed (for PASTA). That type of information is usually useful for planning purposes and for evaluating the performance of a service system, and is performed off-line. Since our main focus is call centers, we would like to mention, that the arrival process to call centers is often non-homogenous, having picks and drops in relatively short intervals. For that reason, typical call centers might not reach a steady state, in its classical form.

The results of the two estimation methods might be substantially different. The following example demonstrates this difference.

Consider an $M/M/s$ system. Recall that in such a system there are s independent and statistically identical servers, service times are exponentially distributed with a mean of $\frac{1}{\mu}$, arrivals are described by a Poisson process (independent of the service process) with a rate of λ, satisfying $\lambda < s \cdot \mu$. Assume that there is no abandonment. The system state at any moment can be described by the number of customers in the system.

i. Given $L + s$ $(L \geq 0)$ customers in the system upon arrival, waiting time is $Erlang(L + 1, s\mu)$. When L is large enough, waiting time can be approximated by a $Normal\left(\frac{L+1}{s\mu}, \frac{L+1}{(s\mu)^2}\right)$ variable.

ii. Without system state information, and assuming a steady state, the waiting time, provided that there is waiting, is exponential $(s \cdot (1 - \rho))$, when $\rho = \frac{\lambda}{s\mu}$.

Individual customers are usually interested in option i.

In this work we study estimations of waiting times for the purpose of informing individuals about their anticipated delays. We therefore focus on estimating times given the system state at the time of estimation (arrival or any point of time during the waiting).

Estimations of waiting times depend on the information provided, and the accuracy of the estimation varies accordingly. For example, when service discipline is FCFS, if we could infer the exact service requirement of each customer upon arrival, we would have been able to anticipate the accurate delay (the system would have become deterministic).

In an ideal situation, we would be able to calculate the full distribution function (conditioned on state) of each customer's waiting time, so that the decision regarding which quantile of that distribution to inform of will be made by the call center manager. As explained before, this decision depends on the desired outcome, and involves marketing and operational considerations. Being informed on long waiting times, customers are more likely to abandon the system and (hopefully) call back later. Being informed on short waiting times, customers will probably choose to wait, but then, as the waiting becomes longer than predicted, dissatisfaction and loss of trust might occur.

However, for complicated models we are sometimes only able to provide less information, such as the mean of the anticipated delay. Whitt [33] demonstrates that when the prediction considers state information, estimations of a single point of the distribution, such as the mean, are often of value. Information regarding system state at the time of estimation tends to make the conditional waiting-time cdf concentrate more about its mean, so that a single point estimate becomes much more reliable, than it is for unconditional estimations.

In practice, with the lack of a good mechanism for estimating waiting times many call center managers provide other queue related information, such as the number of customers already waiting or the elapsed waiting time of the longest-waiting-customer.

As was already mentioned in Chapter 2, queue size information should sometimes be preferred over waiting time estimations (when the expected waiting time is long; see [13]). However, queue size information might be confusing, when customers do not know how many agents provide service nor the overall service times. To overcome this problem, in some call centers [21] the announced queue position already accounts for the number of servers. For example, in a call center with 10 servers, each of the first 10 customers in line can be told that he is the first in queue. On-going updates of the queue state or past experience enable to infer the service rates. Using the queue size information to estimate one's remaining waiting is more challenging when non FCFS service disciplines are implemented.

Another example of measured information is the waiting time of the longest waiting customer. This, in fact, can provide a good estimation of the average delay one is expected to experience in loaded systems and assuming a steady state. Though, this is not always intuitive, it seems correct due to the arrival rate and departure rate being equal. We leave the research of such an estimation for future work.

3.2 Examples of estimating waiting times

We begin with estimations of waiting times when the service discipline is FCFS. Methods of estimating anticipated delays in such systems are studied in Whitt [33], and a large part of our discussion in the current section is based on this paper.

Since the service discipline is FCFS, the waiting time of a customer depends on the number of customers waiting upon his arrival, but not on future arrivals. Since the number of customers already waiting is known, estimations are not affected by the arrival process. The Poisson arrival assumption, often applied in the analysis of queueing systems, may therefore be omitted.

We will now discuss estimation methods (exact calculations and approximations) for the following models:

i. M/M/s and G/M/s

ii. G/M/s with abandonment

iii. When customers are not statistically identical

iv. G/G/s: when service times are not necessarily exponential

3.2.1 M/M/s and G/M/s

As explained in Section 3.1, system state is completely described by the number of customers in the system. The waiting time of a customer, who finds all servers busy and l ($l \geq 0$) customers in queue upon arrival, is $Erlang(l+1,\ s\mu)$ and may be approximated by the Normal distribution when l is large enough.

3.2.2 G/M/s with abandonment

Suppose that each waiting customer in position j of the queue is willing to wait an exponential time with a rate of δ_j. Queue advancements occur due to either service completion or abandonment. The time until the customer in the k'th position in queue will be advanced to the (k-1)'th position is exponential with a rate of $s\mu + \Delta_k$, where $\Delta_k = \sum_{j=1}^{k-1} \delta_j$. The time waited for service of a customer who find $s+l$ customers in the system upon arrival is the sum of $l+1$ independent but not statistically identical exponential. The mean and standard deviation of the anticipated waiting time, W, are:

$$\begin{aligned} E[W] &= \sum_{j=0}^{l} \frac{1}{s\mu + \Delta_j} \\ SD[W] &= \left[\sum_{j=0}^{l} \frac{1}{(s\mu + \Delta_j)^2} \right]^{\frac{1}{2}}, \end{aligned}$$

where

$$\Delta_k = \sum_{j=1}^{k-1} \delta_j. \tag{3.1}$$

The Laplace transform of the waiting time $L_W(\theta)$ is

$$L_W(\theta) = \prod_{j=0}^{l} \frac{s\mu + \Delta_j}{s\mu + \Delta_j + \theta}.$$

Notice, that this analysis allows the patience of a customer to depend on his position in queue. This is reasonable if this information is available to the waiting customer. Otherwise, it is more natural to assume that patience is a function of the elapsed time, or in the simplest case constant. For example, the time that a customer is willing to wait is exponential with a mean of α^{-1}.

3.2.3 When customers are not statistically identical

Often several service types are provided and then customers are not statistically identical. In [33], Whitt distinguishes between cases when customer identity becomes available only upon beginning of service and cases when this information is given already upon arrival. We now discuss methods to estimate waiting times for each of the two cases.

i. **Customer identity is revealed only when service begins:** Consider a system with s statistically identical servers, and two types of service. Service times are exponential with a mean of $\frac{1}{\mu_1}$ for type 1 customers and $\frac{1}{\mu_2}$ for type 2 customers. Assume that each customer is of type 1 with probability p and of type 2 with probability $(1-p)$, independently of other customers. Since the service type of each customer in queue is not known and is observed only when the customer begins service, the service time is hyper-exponential. This is therefore a $G/H_2/s$ model. A recursive algorithm was developed to calculate the waiting time of a customer who finds all servers busy, j type 1 customers being served ($j \leq s$), and l customers waiting in queue [33]. When abandonment is allowed, the time until the first departure is exponential with mean $(j\mu_1 + (s-j)\mu_2 + \Delta_l)^{-1}$, where Δ_l is as defined in (3.1). Let $T(l,j)$ denote the remaining waiting time, not counting the time until the first departure. At the time of the first departure, our customer is advanced by one place in queue. The departure might occur due to abandonment of some other customer (with probability $\frac{\Delta_l}{(j\mu_1+(s-j)\mu_2+\Delta_l)}$), due to a service completion of a type 1 customer (with probability $\frac{j\mu_1}{(j\mu_1+(s-j)\mu_2+\Delta_l)}$), or due to a service completion of a type 2 customer (with probability $\frac{(s-j)\mu_2}{(j\mu_1+(s-j)\mu_2+\Delta_l)}$). If the departure was caused by a service completion, then with probability p the customer enters service is a type 1 customer. The mean waiting time of our customer, $E[W(l,j)]$, is thus given by the following recursion:

$$E[W(l,j)] = \frac{1}{j\mu_1 + (s-j)\mu_2 + \Delta_l} + E[T(l,j)],$$

where for $l \geq 1$,

$$E[T(l,j)] = \frac{p(s-j)\mu_2}{j\mu_1 + (s-j)\mu_2 + \Delta_l} \cdot E[W(l-1,j+1)]$$

$$+\frac{(1-p)j\mu_1}{j\mu_1+(s-j)\mu_2+\Delta_l}\cdot E[W(l-1,j-1)]$$
$$+\frac{pj\mu_1+(1-p)(s-j)\mu_2+\Delta_l}{j\mu_1+(s-j)\mu_2+\Delta_l}\cdot E[W(l-1,j)]$$

and

$$E[T(0,j)]=0.$$

Similarly to the above, recursive formulas can be developed for the variance and for the Laplace transform of the waiting times (see [33] for further details).

ii. **Customer identity is already available upon arrival:** We now assume that customers are classified upon arrival. We still assume that service times are independent of each other and are exponentially distributed, but there may be several service types, so that the mean of exponentials is different. We consider a customer who, upon arrival, finds $s+l$ customers in the system. The vector of $s+l$ individual service rates $(\mu_1,\mu_2,...,\mu_{s+l})$ is known (s customers in service are listed first, and the customers in queue are following them in the waiting order). Abandonment is allowed (exponentially) and the vector of abandonment rates $(\alpha_1,...,\alpha_l)$ is also known. Note that each customer may be willing to wait a different amount of time. This time depends on the customer and not on the position in queue, as was the case in previous model.

The time until first advancement in queue is exponential with a rate of $(\sum_{i=1}^{s}\mu_i+\sum_{i=1}^{l}\alpha_i)$. Times between any successive advancements in queue afterwards (inter-departure times) are also exponential, but their means depend on earlier events. In order to exactly calculate the mean time between successive advancements in queue we need to know who are the customers in service and who are the customers in queue at each stage. In other words, we need to know who are the customers who left the system. Consider, for example, the time between the first advancement in queue of our customer and the second one. Assume that we know that the first advancement in queue appeared due to service completion. In this case, one of the s originally served customers left the system and the (s+1)'th customer entered service. This information does not suffice for calculating the new service rate of the system. Information regarding who is the customer that left is required. If the first advancement in queue appeared due to an abandoning customer (and not due to service completion), then in order to calculate the time until the next advancement, information regarding who is the customer that abandoned is required. One can, of course, use probability based computations, but as the number of servers and queue size increase, computations become long and complicated.

In [33], Whitt develops stochastic upper and lower bounds on the rates of inter-departure times. He then uses these bounds to develop upper and lower bounds on the waiting time distribution. The bounding sets of s service rates and $(l+1-n)$ abandonment rates for the n'th inter-departure time are denoted by $\{\mu_{n,1}, ..., \mu_{n,s}\}$ and $\{\alpha_{n,1}, ..., \alpha_{n,l+1-n}\}$ respectively. For the first inter-departure rate we know the exact rates of service times and of abandonment and they are denoted by $\{\mu_{1,1}, ..., \mu_{1,s}\} \equiv \{\mu_1, ..., \mu_s\}$ and $\{\alpha_{1,1}, ..., \alpha_{1,l+1-n}\} \equiv \{\alpha_1, ..., \alpha_l\}$. The n'th $(1 \leq n \leq l+1)$ upper (lower) bound set $\{\mu_{n,1}, ..., \mu_{n,s}\}$ contains the s smallest (largest) elements from the set $\{\mu_1, ..., \mu_{s+n-1}\}$. The n'th upper (lower) bound set $\{\alpha_{n,1}, ..., \alpha_{n,l+1-n}\}$ contains the $(l+1-n)$ smallest (largest) elements from the set $\{\alpha_1, ..., \alpha_l\}$. The bounding waiting time W_b has the mean:

$$E[W_b] \approx \sum_{n=1}^{l+1} \left[\sum_{i=1}^{l+1-n} \alpha_{n,i} + \sum_{i=1}^{s} \mu_{n,i} \right]^{-1}.$$

Since the means of each exponential variable are bounded, stochastic bounds on the entire waiting time distribution can be obtained:

$$P(W_b^l > t) \leq P(W > t) \leq P(W_b^u > t),$$

where W_b^u and W_b^l are the upper and lower bounds.

3.2.4 G/G/s: Service times are not necessarily exponential

We now consider models with i.i.d. service times with some known distribution. We let T stand for the service time. Assume that there is no abandonment and that the mean service time is $\frac{1}{\mu}$. The mean waiting time of a customer who finds (l) customers in front of him in line can be approximated by:

$$E[W] \approx \frac{l+1}{s\mu}. \tag{3.2}$$

The standard deviation of that waiting time can be approximated by:

$$SD[W] \approx \sqrt{l+1}\frac{SD(T)}{s}. \tag{3.3}$$

The waiting time can be approximated by a *Normal* variable with the mean in (3.2) and the standard deviation in (3.3). This approximation will probably perform better for a smaller s and a larger l.

When service times are not exponential, estimations of waiting times can be improved by estimating the remaining service times of the customers in service. Such estimations can be done based on elapsed service times, number of customers in queue, or simply the distribution of service times. We now elaborate on some possible estimation methods.

i. *Estimating the remaining service times based on the elapsed service times (T^{el}):* it is possible that the service provider keeps track of the starting times for each service in process. At the time of a new arrival, the elapsed service times (ages) of currently served customers are known. The cumulative distribution function of the remaining service time (T^{re}) can be computed. Let the total service time be denoted by T, and let $P\left(T \leq t\right) = G_T(t)$, then $T = T^{re} + T^{el}$ and

$$\begin{aligned} P\left(T^{re} > t | T^{el} = x\right) &= P\left(T > t + x | T > x\right) \\ &= \frac{P(T > t + x)}{P(T > x)} \\ &= \frac{1 - G_T(t + x)}{1 - G_T(x)}, \end{aligned}$$

for $t \geq 0$. Since the function $G_T(t)$ is known, calculation is immediate.

ii. *Estimating the remaining service times based on the number of customers in queue*: when the elapsed service time is not measured, some information on the elapsed service times and on the remaining service times can sometimes be deduced from the number of customers in queue. For example, when no one is waiting, the probability that the current service time began "a long" time ago decreases. We do not present here a full analysis of the method, but simply mention its existence.

iii. *Estimating the remaining service times based on service time distribution:* we can estimate the residual of the service time based on the known distribution of the service time. The result will be more accurate than simply assuming that the remaining service time is distributed as an entire one. For example, the mean of the remaining service time, when there is no information about the elapsed service time, is:

$$E[T^{re}] = \frac{E(T^2)}{E^2(T)}. \tag{3.4}$$

The distributions of remaining service times for customers in service, and of service times for customers in queue, should now be converted into an estimated waiting time.

When there is a single server ($s = 1$), the waiting time is just the sum of the remaining independent service times. For a large number of waiting customers, l, a Normal approximation can be used.

In [33], Whitt suggests an approximation to that waiting time for systems with any number of servers where abandonment is allowed. The waiting time is being expressed in terms of departure process $D(t)$ in the interval $[0, t]$ ($t \geq 0$), where 0 is the time of arrival. The waiting time W is given by:

$$W \;=\; min\{t \geq 0 : \; D(t) = l + 1\}. \tag{3.5}$$

The mean of the waiting time is approximated by approximating $D(t)$ in (3.5) by its mean:

$$E[W] \;\approx\; min\{t > 0 : \; E[D(t)] = l + 1\}. \tag{3.6}$$

The approximation of $D(t)$ by its average in (3.6) can be justified for large waiting times (and a large number of departures) by the law of large numbers. We do not get into further details regarding the approximation and $E[D(t)]$.

3.3 Non FCFS service disciplines

In large call centers, that provide different service types to different customers by many servers, the FCFS model is sometimes inadequate. Different service disciplines are often implemented giving different priorities to customers based on characteristics of the customers (and the service they require), or based on server skills. By priorities we mean that the order of service entries among customers in queue is not determined by their arrival order alone. We say that customers of type 1 have priority over customers of type 2, if a type 1 customer may enter service before a type 2 customer, even when the type 2 customer has been waiting longer. The queue size at the time of arrival, therefore, does not suffice for determining the number of customers that will enter service before the new arrival. Later arrivals may pass him in line if they are of higher priorities.

In this section, we discuss methods of estimating waiting times when the order of service depends on server skills and on customer type. First we introduce the *skills-based routing*: a general model that accounts for both service types and server skills. Then we propose methods for estimating the waiting times in some special cases of the general model, assuming exponential service times and Poisson arrivals. The rest of this section is organized as follows:

i. *Skills-based routing and priority service disciplines:* we explain the principles of skills-based routing and of priority service disciplines and describe a general model.

ii. *A single server and two service types:* we start with a simple priority model with only a single server and two service types. One service type has a priority over the other. We propose a method to estimate waiting times for each type of customers. To do this we use *busy period* logic.

iii. *A single server and n service types:* we elaborate on the two service types model, and estimate waiting times when there is a single server and any number of service types. Service types differ by both their required service times and their priority.

iv. *s servers, service rates depend on the server:* we proceed with estimations of waiting times for systems with any number of servers. For simplicity, we first assume that customers of different types differ only by their priority and not by the required service times. Servers, however, are not necessarily statistically identical.

v. *s servers, service times depend on service types:* we refer to a model with any number of statistically identical servers and with several service types, when service types differ by both their priorities and the required service times. This model is being further discussed later on in this work, and we refer the readers to the relevant chapters.

3.3.1 Skills-based routing and a priority service discipline

Reasons for implementing priorities can be both quantitative (to keep the total waiting time in the system or the total queue size minimal) or qualitative (to maintain higher level of service to "VIP" customers). In large systems, servers are usually not statistically identical. Some of them are faster and better in one task while others are experts in other tasks. Also, in large call centers employees turnover is fast. New employees are usually slower than senior ones. Such differences in server skills justify routing methods which are not FCFS.

Priority service discipline is translated into rules of assigning calls to servers. These rules can be static or depend on system state. When priority depends on system state, assignment rules may change, for example when queue size exceeds a certain value or waiting times of low priority customers become too long. By a static priority we mean that the order of service is predefined according to customer type and to server skills and does not change with the system state. Priorities can also be implemented in a preemptive or a non-preemptive way. When preemptive priority is implemented, service can be interrupted in the middle by an arrival of a customer with a higher priority.

The Skills-Based Routing is an operational method for managing call centers, when taking into account the differences between servers and between customers. Skills-based routing determines the on-line routing rules of customers to servers according to parameters such as: customer type, required service type and server skills or capabilities. Often the distribution nature of a service type will be the same for all servers, but the rates might differ. With the general scheme of skills-based routing, different priority policies may be implemented by different servers. For example, one server gives priority to type 1 customers (over type 2), while another server gives priority to type 2 customers, or can serve only type 2 customers, not being allowed to serve type 1 customers at all. A general description of skills-based routing is presented in Figure 3.1. For a further discussion on skills-based routing see [9].

From practical reasons, we apply some simplifications to the general model. We assume exponential service times with means depending both on service types and server skills, and exponential times to abandonment with a mean that depends on service (customer) type. Arrival process of a type j customers is a Poisson process, and is independent of other arrivals and of service process. Different non-preemptive priorities can be implemented by different servers, but the routing rules and the priorities are static, not depending on system state. Calls with the same priority level are answered according to their arrival order (FCFS).

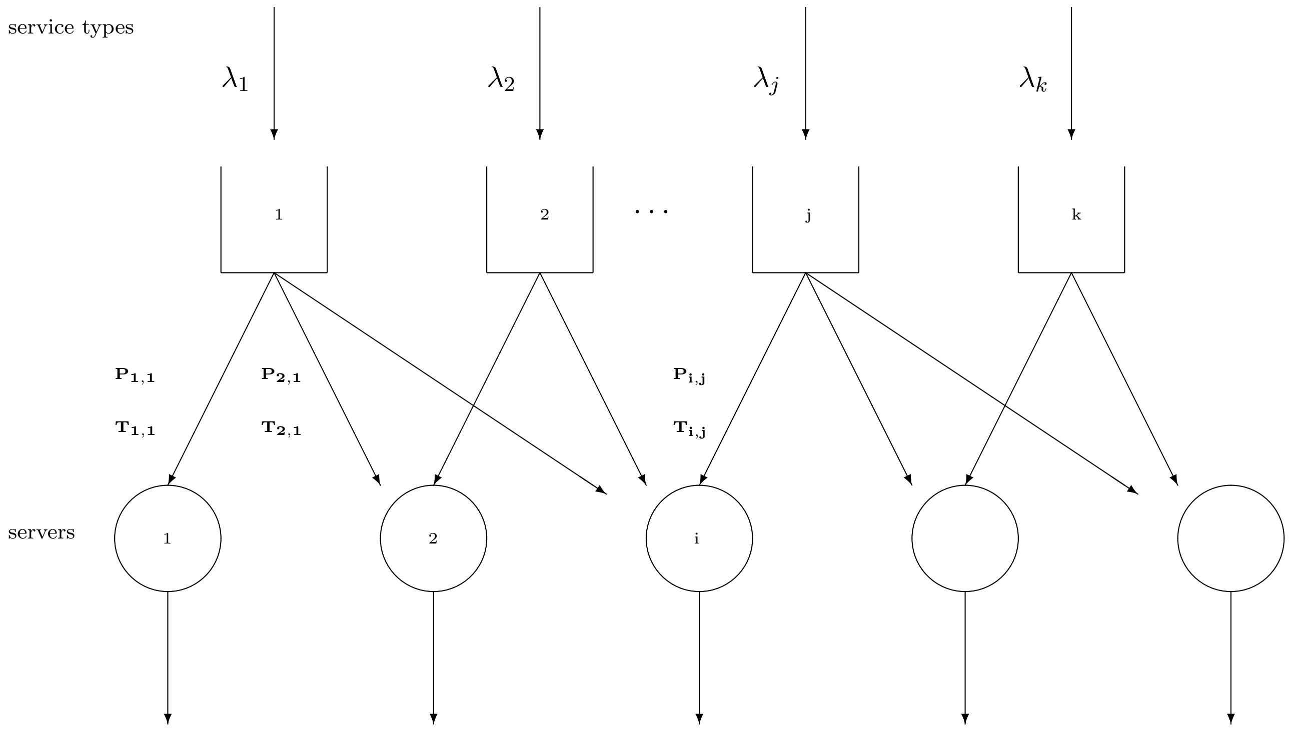
In Chapter 6 we propose an inexplicit analytical approximation for estimations of waiting times in the above model. Meanwhile, we proceed with some additional simplifications.

3.3.2 A single server and two service types

Consider a system with a single server and with two service types: type 1 and type 2. Assume that type 1 has a static non-preemptive priority over type 2, then type 2 customers will not enter service as long as there are type 1 customers in queue. Service times are exponential with means $\frac{1}{\mu_1}$ and $\frac{1}{\mu_2}$ for type 1 and type 2 services respectively. The arrival process of a type j customers ($j = 1, 2$) is a Poisson process with a rate λ_j, and abandonment is not allowed. This is the simplest model for non-preemptive priorities.

System state is described by service configuration and queue configuration, and is denoted by $(S; L_1, L_2)$. $S \in \{0, 1, 2\}$ stands for the service configuration, when $S = 1$ means that a type 1 customer is being served, $S = 2$ if a type 2 customer is being served, and $S = 0$ if the server is free. Queue configuration is denoted by $L = (L_1, L_2)$, where L_1 is the number of type 1 waiting customers and L_2 is the number of type 2 waiting customers. $W_j(S; L_1, L_2)$ stands for the waiting

Figure 3.1: A general scheme of skills-based routing



$\mathbf{T_{i,j}}$ - server's i service time for a type j customer

$\mathbf{P_{i,j}}$ - priority of service type j at server i time of a type j customer ($j = 1, 2$) who finds the system in state $(S; L_1, L_2)$, upon arrival.

Estimations of waiting time are different for type 1 and for type 2 customers. Type 1 customers observe a regular (without priority) queue so estimation of their waiting time is fairly easy. The waiting time of a type 2 customer is affected by future type-1 arrivals. We now demonstrate how waiting times can be estimated for each type of customers.

i. $\mathbf{W_1(S; l_1, \cdot)}$; **Estimating the waiting time of a type 1 customer**
Given $l_1 > 0$ type 1 customers waiting upon arrival, and any number of type 2 customers waiting, the waiting time of a new type-1-arrival consists of:

   - Time until first service completion (one service time of either type 1 or type 2 customer)
   - l_1 independent service times of type 1 customers, which are summed to an $Erlang(l_1, \mu_1)$.

   The two times are independent. Therefore, the moment generating function $M_{W_1(S;l_1,\cdot)}(\theta)$ is

$$M_{W_1(S;l_1,\cdot)}(\theta) \quad = \quad \frac{\mu_S}{\mu_S - \theta} \cdot \left(\frac{\mu_1}{\mu_1 - \theta}\right)^{l_1}. \tag{3.7}$$

ii. $\mathbf{W_2(S; l_1, l_2)}$; **Estimating the waiting time of a type 2 customer**
The waiting time of a type 2 customer who finds l_1 type 1 customers and l_2 type 2 customers waiting upon arrival, can be divided into 3 independent parts:

   - One **busy period** opened by the customer in service.
   - l_1 "standard" busy periods - B - (opened by a type 1 customer).
   - l_2 busy periods opened by a type 2 customer - B_2.

   We will now explain what is a busy period, and how to find its distribution.

Busy period logic

In a system with a single server, we define **B**, a standard busy period, as the time from an arrival of a customer to an empty system until the first time when the system becomes empty again. When there is more than one server, a busy period starts with an arrival to a system with only one idle server (the moment when all servers become busy), and ends at the moment when one of the servers becomes available again, and there is no queue.

In our model of a single server and exponential service times, a busy period can be used for the time since there is only one type 1 customer in the system (served) until there are no type 1 customers, and the server becomes available. A standard busy period, as defined here, does not account for type 2 customers.

We find the moment generating function of a busy period, conditioning on the first step. At the beginning of the period, there is one type 1 customer in service, and no one waits. The system state changes by either an arrival of another type 1 customer (with probability of $\frac{\lambda_1}{\lambda_1+\mu_1}$), or a service completion (with probability $\frac{\mu_1}{\lambda_1+\mu_1}$). If service completion occurs before a new arrival, then the busy period is completed (the busy period in this case consists of one service time only). If a new arrival (of a type 1 customer) occurs during the first service time, then the busy period includes, in addition to the first service time, also the service times of the new arrival and of all future arrivals that will occur during it, etc. Since there is a single server, i.i.d service times, and no abandonment, the service order of the different customers has no impact on the duration of the busy period. In other words, now that there are two customers in the system (one is being served and the other is waiting), we may assume that first all the customers that arrive during the first customer service time, and their "sons" (customers that will arrive during their service times end so on...) will be served. Only when the service of all the customers that belong to the first "arrival tree" is completed, the already waiting customer will enter service. He will then open a new busy period with a new "arrival tree". The result is two independent busy periods: one opened by the originally served customer, and the other opened by the customer who has "just" arrived. Figure 3.2 describes the structure of a busy period.

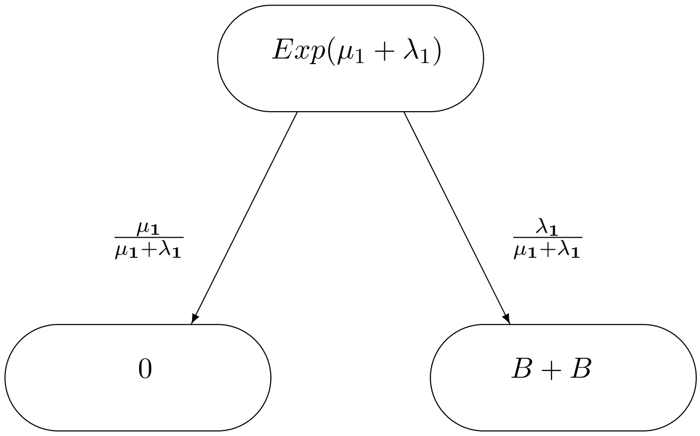
Solving a quadratic equation, the moment generating function of the busy period, $M_B(\theta)$, is found.

$$M_B(\theta) = \frac{\mu_1 + \lambda_1 - \theta - \sqrt{\mu_1^2 - 2\mu_1\lambda_1 - 2\theta\mu_1 + \lambda_1^2 - 2\lambda_1\theta + \theta^2}}{2\lambda_1} \tag{3.8}$$

The average of the busy period is easily differentiated:

$$\begin{aligned} E[B] &= \frac{d}{d\theta} M_B(\theta)|_{\theta=0} \\ &= \frac{1}{\mu_1 - \lambda_1}. \end{aligned} \tag{3.9}$$

Figure 3.2: Structure of a standard busy period


$Exp(\mu_1 + \lambda_1)$
$\frac{\mu_1}{\mu_1+\lambda_1}$
$\frac{\lambda_1}{\mu_1+\lambda_1}$
0
$B + B$

The result in (3.9) becomes intuitive, thinking of a system with a filling rate of λ_1 and an emptiness rate of μ_1 ($\mu_1 > \lambda_1$).

We now define $\mathbf{B}_2$, a non-standard busy period, as a busy period opened by a service of a type 2 customer. It is different from the standard busy period only by the first service. All future arrivals considered are of type 1 customers. The structure of a B_2 busy period is very similar to that of a standard busy period. Its moment generating function is

$$M_{B_2}(\theta) = \frac{\mu_2}{\mu_2 + \lambda_1 - \theta - \lambda_1 M_B(\theta)}, \tag{3.10}$$

and the average of that period is easily computed:

$$E[B_2] = \frac{\mu_1}{\mu_2(\mu_1 - \lambda_1)}. \tag{3.11}$$

We now go back to estimating the waiting time of a low priority (type 2 customer), in a system with a single server. As explained before, that waiting time, denoted by $W_2(S; l_1, l_2)$, is the sum of one busy period opened by the customer in service, l_1 "standard" busy periods (opened by a type 1 customer) and l_2 busy periods of type B_2 (opened by a type 2 customer). All periods are independent of each other. The moment generating function of that waiting time $M_{W_2(S;l_1,l_2)}(\theta)$ is

$$M_{W_2(S;L_1=l_1,L_2=l_2)}(\theta) = \begin{cases} M_B(\theta)^{l_1+1} \cdot M_{B_2}(\theta)^{l_2} & if\ S=1 \\ \\ M_B(\theta)^{l_1} \cdot M_{B_2}(\theta)^{l_2+1} & if\ S=2, \end{cases}$$

when $M_B(\theta)$ is as in (3.8) and $M_{B_2}(\theta)$ is as in (3.10).

3.3.3 A single server and n service types ($\mathbf{n \geq 1}$)

We now allow the system discussed in 3.3.2 to have more than just two service types. Let the number of service types be denoted by n, and assume type 1 customers have the highest priority and type n customers have the lowest priority (all other service types are also ordered according to their priority). Service times of type j customers ($j \in \{1,..,n\}$) are exponentially distributed with a parameter μ_j. Arrival process of type j customers is a Poisson process with a rate λ_j.

Consider a type k customer, who finds (L_1=l_1,....,L_n=l_n) customers waiting, upon arrival, and a type i customer being served. The moment generating function of his waiting time is:

$$M_{W_k(S=i;l_1,...,l_n)}(\theta) \quad = \quad M_{B_i^{(k)}}(\theta) \cdot \prod_{j=1}^{k} [M_{B_j^{(k)}}(\theta)]^{l_j}, \qquad (3.12)$$

where $B_i^{(k)}$ is the busy period opened by a type i customer, as seen by a type k ($k > 1$) customer (meaning, considering all future arrivals of types $(1,..,k-1)$.

The average waiting time of a type k customer can be differentiated from the moment generating function, or found by a recursive formula. Let A_j be number of type j customers that arrive during the waiting time. The average waiting time of a type k customer is:

$$E[W_k(S=i;L_1=l_1,...,L_k=l_k)] \quad = \quad \frac{1}{\mu_i} + \sum_{j=1}^{k} \frac{l_j}{\mu_j} + \sum_{j=1}^{k-1} \frac{E(A_j)}{\mu_j}.$$

Since

$$E(A_j) \quad = \quad \lambda_j \cdot E[W_k(S=i;l_1,...,l_k)],$$

the estimated average time is:

$$E[W_k(S=i;l_1,...,l_k)] \quad = \quad \frac{\frac{1}{\mu_i} + \sum_{j=1}^{k} \frac{l_j}{\mu_j}}{1 - \sum_{j=1}^{k-1} \frac{\lambda_j}{\mu_j}}. \qquad (3.13)$$

In [5], an equivalent result is obtained, by another recursive method.

An explicit expression for the moment generating function of the waiting time in (3.12) can be obtained. First, we denote the moment generating function of a busy period $B_i^{(k)}$, by conditioning on number of arrivals of each type during the first service time. T is the time of the first service. Assume for a moment that this time is known and that $T = t$, hence

$$\begin{aligned} M_{B_i^{(k)}|T}(\theta) &= E[e^{\theta B_i}|T = t] \\ &= e^{\theta \cdot t} \cdot \prod_{j=1}^{k-1} \sum_{a_j=0}^{\infty} \frac{e^{-\lambda_j \cdot t}(\lambda_j t)^{a_j}}{a_j!}[M_{B_j^{(k)}}(\theta)]^{a_j}. \qquad (3.14) \end{aligned}$$

Now we go over all possible values of T and use the fact that $T \overset{d}{=} Exp(\mu_i)$

$$\begin{aligned} M_{B_i^{(k)}}(\theta) &= \int_{t=0}^{\infty} \prod_{j=1}^{k-1} \sum_{a_j=0}^{\infty} \frac{e^{\theta \cdot t} \cdot e^{-\lambda_j t} \cdot [\lambda_j \cdot t \cdot M_{B_j^{(k)}}(\theta)]^{a_j}}{a_j!} f_{S_0}(t)dt \\ &= \int_{t=0}^{\infty} \prod_{j=1}^{k-1} e^{\theta \cdot t} \cdot e^{-\lambda_j t} \cdot e^{\lambda_j \cdot t \cdot M_{B_j^{(k)}}(\theta)} \cdot f_{S_0}(t)dt \\ &= \int_{t=0}^{\infty} e^{\theta \cdot t} \cdot e^{-t \sum_{j=1}^{k-1} \lambda_j} \cdot e^{t \sum_{j=1}^{k-1} \lambda_j M_{B_j^{(k)}}(\theta)} \cdot f_{S_0}(t)\ dt \\ &= M_{S_0}(\theta - \sum_{j=1}^{k-1} \lambda_j + \sum_{j=1}^{k-1} \lambda_j M_{B_j^{(k)}(\theta)}) \\ &= \frac{\mu_i}{\mu_i - \theta + \sum_{j=1}^{k-1} \lambda_j - \sum_{j=1}^{k-1} \lambda_j M_{B_j^{(k)}}(\theta)}. \qquad (3.15) \end{aligned}$$

In this way we get a system of n equations for $M_{B_1^{(k)}}(\theta), ..., M_{B_n^{(k)}}(\theta)$ (for $k > 1$). For a type 1 customer ($k = 1$) the busy period consists of the first service time only. The generating moment of a busy period opened by a type i customer, as it is experienced by a type 1 customer $B_i^{(1)}$is simply

$$M_{B_i^{(1)}}(\theta) = \frac{\mu_i}{\mu_i - \theta}. \qquad (3.16)$$

The results of (3.15) and (3.16) can be used in (3.12) to formulate the generating moment function for the total waiting time.

3.3.4 s servers, service rates depend on the server

Now we consider a system with s independent servers, and n service types. We assume that though there are several service types, they do not differ by their service requirements (for example, "VIP" customers and "regular" customers that need the same service). All customers are independent. However, different servers have different service rates, so that service times at server i ($i \in \{1,..,s\}$) are exponential with a mean $\frac{1}{\mu_i}$. The same non-preemptive priority discipline is implemented by all servers. Services 1 to n are ordered by their priority level, when type 1 has the highest priority level and type n has the lowest priority. Arrival process of type j ($j \in \{1,..,n\}$) customers is a Poisson process with a rate λ_j, and no abandonment is allowed. This model is an extension of the one in Subsection 3.3.3.

When all the servers are busy (which is the relevant situation for our purpose), the time until the first service completion is exponential with a rate of $\hat{\mu} = \sum_{i=1}^{s} \mu_i$. Note that for our purpose it is not important what type of customer is being served by each server (since service times depend on server and not on service type). The waiting time of a type j ($j \geq 1$) customer who finds, upon arrival, all the servers busy and $L = (l_1, ..., l_n)$ customers waiting consists of i.i.d exponential times with a rate $\hat{\mu}$, each corresponds to the time between successive service completions. The number of service completions that our customer should wait is the sum of the number of customers of types $\leq j$ that were waiting before him (known), and the number of arrivals of customers of types $< j$ (unknown), plus 1. Since all customers have the same service requirements, we can look at the accumulated arrival process of types $< j$ rather than at the arrival process of each service type separately. This is a Poisson process with a rate $\hat{\lambda} = \sum_{i=1}^{j-1} \lambda_i$.

The problem can, hence, be reduced into the single server model and two service types that was solved in Subsection 3.3.2. The waiting time of the type j customer who finds, upon arrival, $L = (l_1, ..., l_n)$ customers waiting consists of $(1 + \sum_{i=1}^{j} l_i)$ standard busy periods, in which service times are exponential with a rate $(\hat{\mu})$, and arrival process is Poisson with a rate $(\hat{\lambda})$.

3.3.5 s servers, service times depend on service types

Estimation of waiting times when there are more than one server, more than one service type and when different service types require different service times and have different priorities, becomes complicated. In the next chapter, we focus on estimations of waiting times for simplest case of this model: two servers and two service types. We use difference equations, and explain the difficulties in the exact calculation. Then in Chapter 5, we present another method for the exact calculation of waiting times: matrix geometric technique, and explain the difficulties in that technique as well. We propose an alternative in Chapter 6. In Chapter 9, Section 9.2 we return to the general model and suggest an additional approximation for large systems.

Chapter 4

Exact Analysis: Difference Equations of a 'Simple Case'

In this chapter we describe a specific and relatively simple case for estimating the waiting time in a system with priorities, and in which service times depend on the customers. The special case discussed in this chapter has the following characteristics:

- Two service types: the arrival processes of type 1 and of type 2 customers are time-homogeneous Poisson processes with rates λ_1, λ_2 respectively.
- Service times are exponentially distributed with rates μ_1, μ_2 respectively.
- Type 1 has a static non-preemptive priority over type 2.
- Two servers, statistically identical.
- No abandonment.

The system is analyzed in detail, and the waiting time distribution described by difference equations. For some cases, we also solve the equations and present the explicit solutions. Doing this, we demonstrate the difficulty in achieving an explicit solution for the problem. Later in this work, we present two alternative methods for estimations of waiting times.

- *Matrix geometric solutions*: a technique for analyzing processes a repetitive form.
- *Inexplicit analytical approximation:* an iterative process for approximating waiting times.

We shall use the system analyzed in this chapter to demonstrate the alternative methods. We will refer to this system as the 'simple case'.

4.1 System description and marking convention

Formally, the system state at any given time will be described by:

- $\mathbf{S} = (\mathbf{S_1}, \mathbf{S_2})$ is the service configuration. S_j $(j = 1, 2)$ is the number of type j customers served at a given moment. Since there are only two servers, $S_j \in \{0, 1, 2\}$.
- $\mathbf{L} = (\mathbf{L_1}, \mathbf{L_2})$ is the queue configuration. L_j $(j = 1, 2)$ is the number of type j customers waiting at a given moment.

Since both service times and times between successive arrivals are exponential, and since the two servers are statistically identical, the system state is completely described by the number of customers of each type that are waiting and the number of customers (of each type) that are being served. Information regarding how long customers are waiting or how long they have been served, is irrelevant for our purpose.

Sometimes we would like to describe the system at a specific moment. We will then use $(\textit{system state})^{moment}$ to indicate it. For example: L_1^t represents the number of type 1 waiting customers at moment t. The parameter t is not necessarily continuous and is often used to count iterations or steps. When we describe the system state upon arrival, we will set $t = 0$. For example, S^0 represents the service configuration at the time of arrival.

- $\mathbf{W_j}(\mathbf{S} = (\mathbf{s_1}, \mathbf{s_2}); \mathbf{L} = (\mathbf{l_1}, \mathbf{l_2}))$ is the waiting time of a type j customer $(j = 1, 2)$, who finds the system in state $(S = (s_1, s_2); L = (l_1, l_2))$ at the time of arrival. Sometimes we use the equivalent notation $\mathbf{W_j}(\mathbf{s_1}, \mathbf{s_2}; \mathbf{l_1}, \mathbf{l_2})$, for convenience.

4.2 Detailed analysis - difference equations

In the following sub-sections we discuss calculations of the waiting time distribution of a customer who, upon arrival, finds the system at a given state. The calculations are substantially different for two cases:

i. Waiting time calculations for high priority customers (type 1 arrival)

ii. Waiting time calculations for low priority customers (type 2 arrival)

The waiting time distributions of both type 1 and type 2 customers depend on the service configuration at the time of arrival S^0. There are 4 possible cases:

i. $\mathbf{S^0 = (s_1, s_2)}$ where $s_1 + s_2 < 2$; idle servers.

ii. $\mathbf{S^0 = (2, 0)}$; there are two type 1 customers served upon arrival.

iii. $\mathbf{S^0 = (1, 1)}$; one customer of each type is being served.

iv. $\mathbf{S^0 = (0, 2)}$; two type 2 customers are being served.

Each of the above 4 cases will be analyzed separately. That will be done both for type 1 and for type 2 customers.

4.2.1 Waiting time for a high priority customer

Consider a type 1 customer who finds upon arrival $L^0 = (l_1, l_2)$ customers waiting and $S^0 = (s_1, s_2)$ service configuration. Since this customer will enter service before all the type 2 customers (even if they have arrived before him), his waiting time is independent of L_2. We are, therefore, interested in the waiting time for any value of L_2: $W_1(S = (s_1, s_2); L = (l_1, \cdot))$
We now analyze each of the 4 cases of service configuration.

i. $\mathbf{S^0 = (s_1, s_2)}$ where $s_1 + s_2 < 2$; idle servers

This case is immediate - the customer will not wait. One of the two servers is free only if there are no customers in queue. Thus, the arriving customer immediately enters service. Formally:

$$W_1(0, 0; 0, 0) \overset{d}{=} W_1(1, 0; 0, 0) \overset{d}{=} W_1(0, 1; 0, 0) \equiv 0$$

.

ii. $\mathbf{S^0 = (2, 0)}$; Both served customers are of type 1

In this case, all the customers that are served before our customer enters service are type 1 customers. The waiting time is therefore the sum of all times between queue advancements. Each time a service is completed, one customer (served) leaves the system, another (type 1) customer enters service, and our customer is advanced by one place in queue. Since only type 1 customers are served, the time between any successive advancements in the queue is identically distributed ($Exp(2\mu_1)$), independently of the others.

The total waiting time in this case has an Erlang distribution with the parameters $(l_1 + 1, 2\mu_1)$. Formally:

$$W_1(S = (2, 0); L = (l_1, \cdot)) \overset{d}{=} Erlang(l_1 + 1, 2\mu_1)$$

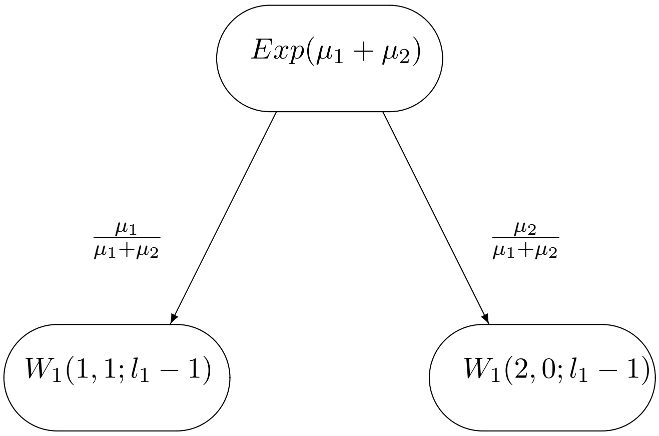
The anticipated mean of the waiting time and its variance are now immediate:

$$\begin{aligned} E\left[W_1(S=(2,0);L=(l_1,\cdot))\right] &= \frac{l_1+1}{2\cdot\mu_1} \\ Var\left[W_1(S=(2,0);L=(l_1,\cdot))\right] &= \frac{l_1+1}{4\cdot\mu_1^2}. \end{aligned}$$

iii. $\mathbf{S^0=(1,1)}$; One customer of each type is being served upon arrival

The waiting time $\mathbf{W_1(S=(1,1);L=(l_1,\cdot))}$ when $l_1 \geq 0$ can be divided into two periods, conditioning on the first service completion, as described in Figure 4.1.

Figure 4.1: Waiting time of a type 1 customer, $S^0=(1,1)$



- The first period is the time until the first service completion. As a minimum between independent exponentials, this time has an $Exp(\mu_1+\mu_2)$ distribution.

- The second period is the remaining time: from the first service completion and until our customer enters service. This time is equal to the waiting time of a customer, who finds the system in the same state as right after first queue advancement. We will let the service configuration after the i'th queue advancement be denoted by S^i.

If the service completion is that of the type 2 customer (with probability $\frac{\mu_2}{\mu_1+\mu_2}$), then the service configuration right after the first queue advancement will be $S^1 = (2,0)$. From this point on (until our waiting customer enters service) only type 1 customers will be served. There are now $(l_1 - 1)$ type 1 customers waiting before our customer, and therefore he will have to wait an additional time with an Erlang $(l_1, 2\mu_1)$ distribution, as explained in the previous section.

If the service of the type 1 customer is completed before the service of type 2 customer, then after the first queue advancement, there will be again one customer of each type served ($S^1 = (1,1)$). The number of type 1 waiting customers is decreased by 1, and therefore our customer will have to wait an additional time, which is equivalent to the time one should wait when he finds the system with $(l_1 - 1)$ type 1 customers waiting (and one customer of each type being served), i.e. $W_1(S = (1,1); L = (l_1 - 1, \cdot))$.

We now find the moment generating function of the waiting time.

Let

$$\mathbf{q} = \frac{\boldsymbol{\mu_1}}{\boldsymbol{\mu_1} + \boldsymbol{\mu_2}} \tag{4.1}$$

The waiting time is presented by the following difference equation:

$$W_1\left(S = (1,1); L = (l_1, \cdot)\right) \overset{d}{=} \begin{cases} T^0 + W_1(1,1; l_1 - 1, \cdot), & w.p \quad q \\ \\ T^0 + W_1(2,0; l_1 - 1, \cdot), & w.p \quad 1-q \end{cases}$$

Where:

- $\mathbf{T^0}$ is the time until the first service completion; $T^0 \overset{d}{=} Exp(\mu_1 + \mu_2)$.
- $\mathbf{W_1(2,0;l_1 - 1}, \cdot)$ has an $Erlang(l_1, 2\mu_1)$ distribution, as was already explained.

Let the moment generating function of a variable W with an argument θ be denoted by $M_W(\theta)$. The moment generating function of $W_1(S = (1,1); L = (l, \cdot))$ is, hence, denoted by $M_{W_1(1,1;\ l,\cdot)}(\theta)$. For convenience, we denote it by $M(l)$. (Note that we now use l instead of l_1 for the number of type 1 waiting customers). Formally:

$$M_{W_1(1,1;\ l,\cdot)}(\theta) = E(e^{\theta W_1(1,1;\ l,\cdot)}) \equiv M(l) \tag{4.2}$$

Since the two time periods are independent,

$$\begin{aligned} M(l) &= \frac{\mu_1+\mu_2}{\mu_1+\mu_2-\theta} \cdot \left[\frac{\mu_1}{\mu_1+\mu_2} \cdot M(l-1) + \frac{\mu_2}{\mu_1+\mu_2} \left(\frac{2\mu_1}{2\mu_1-\theta} \right)^l \right] \\ &= \frac{\mu_1}{\mu_1+\mu_2-\theta} \cdot M(l-1) + \frac{\mu_2}{\mu_1+\mu_2-\theta} \cdot (\frac{2\mu_1}{2\mu_1-\theta})^l. \end{aligned} \tag{4.3}$$

Initial conditions are given by one of the following (equivalents):

$$\begin{aligned} M(-1) &= E(e^{\theta \cdot 0}) = 1 \\ M(0) &= \frac{\mu_1+\mu_2}{\mu_1+\mu_2-\theta} \end{aligned} \tag{4.4}$$

$M(0)$ relates to the waiting time of a customer who finds two customers in service but no one in line. $l = -1$ is an artificial state, representing the trivial case of less than two customers being served. Though the real case is analyzed separately, we can still use $M(-1)$ as an initial condition.

The solution for (4.3), under the above condition (4.4) is:

$$M(l) = (\frac{\mu_1}{\mu_1+\mu_2-\theta})^{l+1} + \sum_{i=0}^{l} (\frac{\mu_1}{\mu_1+\mu_2-\theta})^i \cdot \frac{\mu_2}{\mu_1+\mu_2-\theta} \cdot (\frac{2\mu_1}{2\mu_1-\theta})^{l-i},$$

which simplifies to:

$$\begin{aligned} M(l) = &(\frac{\mu_1}{\mu_1+\mu_2-\theta})^{l+1} \\ &+ \frac{2\mu_2}{2\mu_2-\theta} \cdot (\frac{2\mu_1}{2\mu_1-\theta})^l \cdot \left[1 - (\frac{2\mu_1-\theta}{2\mu_1+2\mu 2-2\theta})^{l+1} \right]. \end{aligned} \tag{4.5}$$

From the moment generating function it is easy to derive the moments of the waiting time. For example, the mean of the expected waiting time is given by:

$$E[W_1(1,1;l,\cdot)] = \frac{d}{d\theta} M_{W_1(1,1;\ l,\cdot)}(\theta)|_{\theta=0}.$$

Differentiating $M(l)$ in (4.5), and letting $\theta = 0$, we obtain:

$$E[W_1(1,1;l,\cdot)] = (\frac{\mu_1}{\mu_1+\mu_2})^l \cdot \frac{\mu_2-\mu_1}{2 \cdot \mu_2 \cdot (\mu_1+\mu_2)} + l \cdot \frac{1}{2 \cdot \mu_1} + \frac{1}{2 \cdot \mu_2}. \tag{4.6}$$

iv. $\mathbf{S^0 = (0, 2)}$; Both served customers are of type 2

The waiting time of a type 1 customer who finds the system in the state $S^0 = (0, 2)$ and $L^0 = (l, \cdot)$ consists of two periods:

- The time until the first customer leaves service, which is exponentially distributed with a rate of $(2\mu_2)$.
- Additional time, distributed the same as $W_1(1, 1; l-1, \cdot)$. This time was already analyzed.

Since the two time periods are independent, the moment generating function is:

$$M_{W_1(0,2;\ l,\cdot)}(\theta) \;=\; \frac{2\mu_2}{2\mu_2 - \theta} \cdot M_{W_1(1,1;\ l-1,\cdot)}(\theta), \tag{4.7}$$

where $M_{W_1(1,1;l-1,\cdot)}(\theta)$ is given in (4.5). All moments of the waiting time can be easily found now. For example, to calculate the mean:

$$E[W_1(S = (0, 2); L = (l, \cdot)] \;=\; \frac{1}{2\mu_2} + E[W_1(S = (1, 1); L = (l-1, \cdot))].$$

$E[W_1(S = (1, 1); L = (l-1, \cdot))]$ is given in (4.6). That implies:

$$E[W_1(0, 2; l, \cdot)] \;=\; (\frac{\mu_1}{\mu_1 + \mu_2})^l \cdot \frac{\mu_2 - \mu_1}{2\mu_1\mu_2} + (l-1) \cdot \frac{1}{2\mu_1} + \frac{1}{\mu_2}. \tag{4.8}$$

4.2.2 Waiting time of a low priority customer (type 2), when $L_2^0 = 0$

First we analyze the case in which our customer finds no type 2 customers in line, upon arrival. Then, we use the result to analyze the more general case.

Again, there are 4 cases which differ from each other by the initial service configuration:

i. $\mathbf{S^0 = (s_1, s_2)}$, where $s_1 + s_2 < 2$; idle servers upon arrival.

ii. $\mathbf{S^0 = (2, 0)}$; two type 1 customers are being served upon arrival.

iii. $\mathbf{S^0 = (1, 1)}$; one customer of each type is being served upon arrival.

iv. $\mathbf{S^0 = (0, 2)}$; two type 2 customers are being served upon arrival.

i. $\mathbf{S^0 = (s_1, s_2)}$, where $\mathbf{s_1 + s_2 < 2}$

This case is immediate, since the customer will not wait:

$$W_2(0,0;0,0) \overset{d}{=} W_2(1,0;0,0) \overset{d}{=} W_2(0,1;0,0) \equiv 0.$$

ii. $\mathbf{S^0 = (2,0)}$; Both served customers are of type 1

We denote the moment generating function of the waiting time $W_2(S=(2,0); L=(l,0))$ by $f(l)$. Formally:

$$M_{W_2(2,0;\ l,0)}(\theta) \quad \equiv \quad f(l).$$

Thus, using the same methodology as in the previous analysis:

$$f(l) \quad = \quad \frac{2\mu_1 + \lambda_1}{2\mu_1 + \lambda_1 - \theta} \cdot [\frac{\lambda_1}{2\mu_1 + \lambda_1} f(l+1) + \frac{2\mu_1}{2\mu_1 + \lambda_1} f(l-1)],$$

which simplifies to:

$$f(l) \quad = \quad \frac{2\mu_1}{2\mu_1 + \lambda_1 - t} \cdot f(l-1) + \frac{\lambda_1}{2\mu_1 + \lambda_1 - t} \cdot f(l+1). \tag{4.9}$$

The initial condition is:

$$f(-1) \quad = \quad 1. \tag{4.10}$$

To solve the second order difference equation (4.9), we need another initial or boundary condition. We use the *busy period logic* to find $f(0)$, similarly to the explanation regarding busy period in Subsection 3.3.2.
$f(0)$ is the moment generating function of the waiting time of a type 2 customer, who finds two type 1 customers in service and no customers in queue. That waiting time can be divided as follows:

- $\mathbf{T^0}$: time until first service completion (or: first "**iteration**" time), which is $Exp(2\mu_1)$.
- **N** additional periods, each distributed as $W_2(2,0;0,0)$, where N is the number of type 1 customers, who arrived during the first iteration.

The service order of type 1 customers does not impact the waiting time. Therefore, we may refer to the situation as if each of the N customers opens an independent busy period. Let B stand for a busy period, and $M_B(\theta)$ for its generating moment function.

$$f(0) = M_B(\theta) = E_N \left[\frac{2\mu_1}{2\mu_1 - \theta} [M_B(\theta)]^N \right] \tag{4.11}$$

The moment generating function of the busy period B is given by:

$$\begin{aligned} M_B(\theta) &= \int_{t=0}^{\infty} \sum_{n=0}^{\infty} \frac{(\lambda t)^n}{n!} e^{-\lambda t} \cdot e^{\theta s} M_B^n(\theta) f_T(t) dt \\ &= \int_{t=0}^{\infty} e^{t(\lambda M_B(\theta) - \lambda + \theta)} f_T(t) dt \\ &= M_T(\lambda M_B(\theta) - \lambda + \theta). \end{aligned} \tag{4.12}$$

Using the fact that $T \overset{d}{=} Exp(2\mu_1)$, we can explicitly express $M_T(\lambda M_B(\theta) - \lambda + \theta)$ in (4.12), and obtain a quadratic equation for $M_B(\theta)$:

$$f(0) = M_B(\theta) = \frac{2\mu_1}{2\mu_1 - \lambda_1 M_B(\theta) + \lambda_1 - \theta} \tag{4.13}$$

The solution for the above equation is:

$$M_B(\theta) = \frac{-(2\mu_1 + \lambda_1 - \theta) \pm \sqrt{(2\mu_1 + \lambda_1 - \theta)^2 - 8\lambda_1\mu_1}}{-2\lambda_1} \tag{4.14}$$

To calculate the average waiting time of a type 2 customer, who finds two type 1 customers in service and no waiting customers, we first differentiate f(0) and then let $\theta = 0$:

$$\frac{d}{d\theta} f(0) = -\frac{1}{2\lambda_1} \pm \frac{1}{2\lambda_1} \cdot \frac{2\mu_1 + \lambda_1 - \theta}{\sqrt{(2\mu_1 + \lambda_1}} \cdot (-1)$$

Now letting $\theta = 0$ we obtain:

$$\frac{d}{d\theta} f(0)|_{\theta=0} = E[W_2(2,0;0,0)] = \begin{cases} -\frac{2\mu_1}{\lambda_1(2\mu_1 - \lambda_1)} \\ \frac{1}{2\mu_1 - \lambda_1} \end{cases} \tag{4.15}$$

Since $\frac{d}{d\theta} f(0)|_{\theta=0}$ should have a positive value, and under the assumption that $2\mu_1 - \lambda_1 > 0$, we conclude that:

$$f(0) = M_B(\theta) = \frac{(2\mu_1 + \lambda_1 - \theta) - \sqrt{(2\mu_1 + \lambda_1 - \theta)^2 - 8\lambda_1\mu_1}}{2\lambda_1}. \tag{4.16}$$

The average waiting time of a type 2 customer, who finds no customers in queue and two type 1 customers in service, is, therefore, $\frac{1}{2\mu_1 - \lambda_1}$. This result may be perfectly understood by thinking of a system with a demand rate of λ_1 and emptiness rate of $2\mu_1$ and looking for the average time until the system becomes "empty".

$f(0)$ as given in (4.16) is the second initial condition for (4.9), the difference equation for $f(l)$.

The solution for (4.9) under the conditions in (4.10) and in (4.16) is given by:

$$\mathbf{f(l) = f(0)^{l+1}} \tag{4.17}$$

The above results can be summarized by:

$$M_{W_2(2,0;\ l,0)}(\theta) = \left(\frac{(2\mu_1 + \lambda_1 - \theta) - \sqrt{(2\mu_1 + \lambda_1 - \theta)^2 - 8\lambda_1\mu_1}}{2\lambda_1} \right)^{l+1} \tag{4.18}$$

This result becomes intuitive, if we look at each of the l waiting customers as opening a new busy period, so that we have $(l+1)$ independent statistically identical busy periods.

We could have also refer to this case as a M/M/1 system (with the same priority policy). This is enabled by the fact that all iterations have the same time distribution ($Exp(2\mu_1)$).

iii. $\mathbf{S^0 = (1,1)}$; One customer of each type is being served

The waiting time $W_2(1,1;l_1,0)$ is divided into two stages, as described in Figure 4.2.

We define $g(l)$ as the moment generating function $M_{W_2(1,1,l,0)}(\theta)$.

$$M_{W_2(1,1,l,0)}(\theta) \equiv g(l). \tag{4.19}$$

Hence,

$$\begin{aligned} g(l) &= \frac{\mu_1 + \mu_2 + \lambda_1}{\mu_1 + \mu_2 + \lambda_1 - \theta} \left[\frac{\mu_1}{\mu_1 + \mu_2 + \lambda_1} g(l-1) + \frac{\mu_2}{\mu_1 + \mu_2 + \lambda_1} f(l-1) + \frac{\lambda_1}{\mu_1 + \mu_2 + \lambda_1} g(l+1) \right] \\ &= \frac{\mu_1}{\mu_1 + \mu_2 + \lambda_1 - \theta} \cdot g(l-1) + \frac{\lambda_1}{\mu_1 + \mu_2 + \lambda_1 - \theta} \cdot g(l+1) \\ &\quad + \frac{\mu_2}{\mu_1 + \mu_2 + \lambda_1 - \theta} \cdot f(l-1). \end{aligned} \tag{4.20}$$

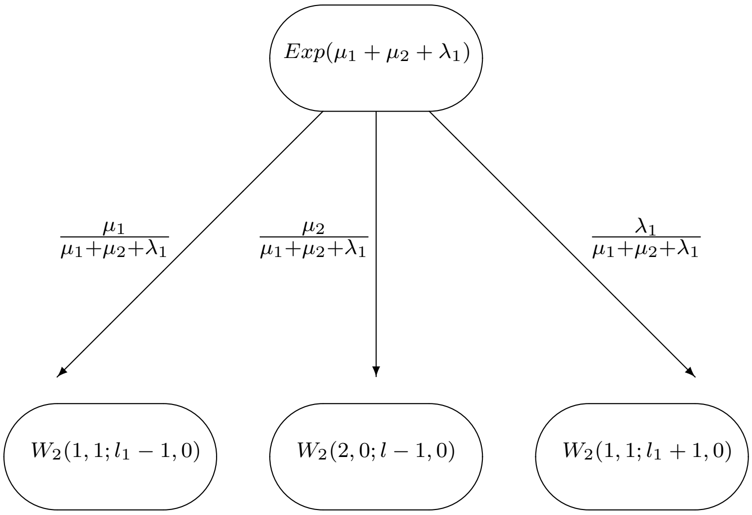
$f(l)$ was found in the previous section and is given in (4.17).

Initial and boundary conditions for $\mathbf{g(l)}$

i. A customer who finds an available server will not wait: $g(-1) = 1$.

ii. The total waiting time may be divided into $(l+1)$ independent busy periods:

Figure 4.2: Type 2 customer's waiting time, $S^0 = (1, 1)$, $L^0 = (l_1, 0)$


$Exp(\mu_1 + \mu_2 + \lambda_1)$
$\frac{\mu_1}{\mu_1+\mu_2+\lambda_1}$
$\frac{\mu_2}{\mu_1+\mu_2+\lambda_1}$
$\frac{\lambda_1}{\mu_1+\mu_2+\lambda_1}$
$W_2(1, 1; l_1 - 1, 0)$
$W_2(2, 0; l - 1, 0)$
$W_2(1, 1; l_1 + 1, 0)$

- The first busy period starts with arrival and ends after all type 1 customers who arrived during it, enter service. That happens just before the first originally waiting type 1 customer enters service.
- Each of the remaining l busy periods is opened by one of the l customers in line.

This is correct only because the order of serving type 1 customers does not matter.

While the service configuration at the beginning of the first busy period is known, $S^0 = (1, 1)$, the service configuration at the beginning of each of the following busy period depends on which of the customers (type 1 or type 2) is the first to leave service.
We will now look into service configuration probabilities at the moment when the first busy period is over, and the first type 1 customer who was originally in queue enters service. It is possible that in service there will be two type 1 customers (if the type 2 customer who was served in the initial state has already left), or there will be, again, one customer of each type in service (if the type 2 customer has not left yet).
Let **r** be the probability that at the end of the first busy period, the type 2

customer will still be in service.

$$r \quad = \quad P\left(S^{\overset{end\ of}{busy\ period}} = (1,1) | S^{\overset{start\ of}{busy\ period}} = (1,1)\right) \tag{4.21}$$

Recall that $S^0 = (s_1, s_2)$ is the service configuration at the beginning of the period. $S^B = (S_1^B, S_2^B)$ stands for the service configuration right after the end of the period, and at the beginning of the next busy period.
We can find r by conditioning on the first step:

$$\begin{aligned} r &= P\left(S^B = (1,1)|S^0 = (1,1)\right) \\ &= \frac{\mu_1}{\mu_1+\mu_2+\lambda_1}\cdot 1 + \frac{\mu_2}{\mu_1+\mu_2+\lambda_1}\cdot 0 + \frac{\lambda_1}{\mu_1+\mu_2+\lambda_1}\cdot r^2 \\ &= \frac{\mu_1}{\mu_1+\mu_2+\lambda_1} + \frac{\lambda_1}{\mu_1+\mu_2+\lambda_1} r^2. \end{aligned}$$

The solution of the above quadratic equation is:

$$r_{1,2} \quad = \quad \frac{\mu_1+\mu_2+\lambda_1 \pm \sqrt{(\mu_1+\mu_2+\lambda_1)^2 - 4\lambda_1\mu_1}}{2\lambda_1}.$$

r represents a probability, so it should be in the range $0 \le r \le 1$. Also, $\mu_1,\ \mu_2,\ \lambda_1\ >\ 0$. We thus obtain the exact value of r:

$$r = \frac{\mu_1+\mu_2+\lambda_1 - \sqrt{(\mu_1+\mu_2+\lambda_1)^2 - 4\lambda_1\mu_1}}{2\lambda_1}. \tag{4.22}$$

Now we can write the second condition for the difference equations:

$$g(l) \quad = \quad g(0)\left[(1-r)\cdot f(l-1) + r\cdot g(l-1)\right], \tag{4.23}$$

or

$$g(1) \quad = \quad g(0)\left[(1-r)\cdot f(0) + r\cdot g(0)\right].$$

***The solution for* g(l)**

The solution for the homogeneous equation is of the form:

$$\begin{aligned} g^h(l) \quad = \quad & C_1 \cdot \left[\frac{\mu_1+\mu_2+\lambda_1-\theta+\sqrt{(\mu_1+\mu_2+\lambda_1-\theta)^2-4\lambda_1\mu_1}}{2\lambda_1}\right]^l \\ & + C_2 \cdot \left[\frac{\mu_1+\mu_2+\lambda_1-\theta-\sqrt{(\mu_1+\mu_2+\lambda_1-\theta)^2-4\lambda_1\mu_1}}{2\lambda_1}\right]^l, \end{aligned}$$

where C_1 and C_2 can be found according to initial and boundary conditions. The solution for the non-homogeneous solution is:

$$g(l) \quad = \quad A \cdot f^l(0),$$

where

$$A = \frac{\mu_2}{\lambda_1 f^2(0) - (\mu_1 + \mu_2 + \lambda_1 - \theta) \cdot f(0) + \mu_1} \cdot f(0). \tag{4.24}$$

Additional information about the average waiting time when there are no customers in line

Even without the explicit solution for $g(l)$, we can still explicitly express the average waiting time when there are no customers in line, $E\left[W_2(1,1;0,0)\right]$. $\mathbf{A_1(1,1)}$ is the number of type 1 customers that arrive during the busy period (started with a service configuration of $S = (1,1)$). Hence:

$$E\left[W_2(1,1;0,0)\right] \quad = \quad E_{A_1(1,1)} \sum_{i=0}^{A_1(1,1)} \left[\frac{q^i}{\mu_1 + \mu_2} + \frac{1 - q^i}{2\mu_1}\right]$$

($q = \frac{\mu_1}{\mu_1 + \mu_2}$ as defined in equation (4.1)). From the above we may deduce that:

$$\begin{aligned} & E\left[W_2(1,1;0,0)\right] \cdot \left(\frac{2\mu_1 - \lambda_1}{2\mu_1}\right) \\ = \quad & \frac{1 - Eq^{A_1(1,1)+1}}{\mu_2} + \frac{1}{2\mu_1} - \frac{1 - Eq^{A_1(1,1)+1}}{2\mu_1\mu_2}(\mu_1 + \mu_2) \end{aligned} \tag{4.25}$$

Since:

$$P\Big(S^B = (1,1)|S^0 = (1,1), A_1(1,1) = a\Big) = q^{a+1},$$

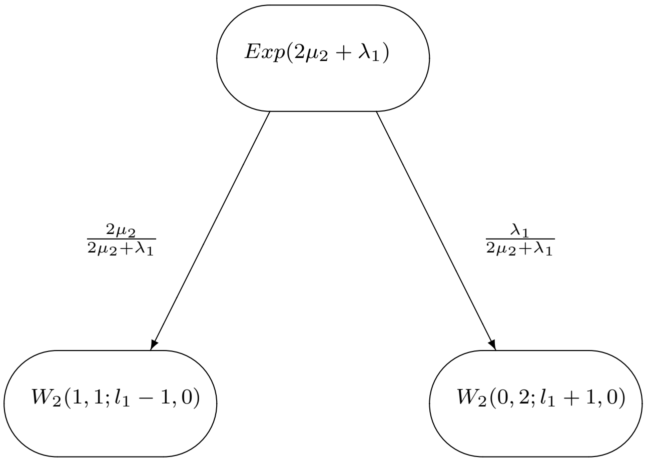
we may say that:

$$P(S^B = (1,1)|S^0 = (1,1)) = E(q^{A_1(1,1)+1})$$

Note, that $P(S^B = (1,1)|S^0 = (1,1))$ is equivalent to r, which was found in (4.22). This relation is now used to conclude that:

$$\begin{aligned} E\left[W_2(1,1;0,0)\right] \cdot \left(\frac{2\mu_1 - \lambda_1}{2\mu_1}\right) \quad &= \quad \frac{dg(0)}{d\theta}|_{\theta=0} \cdot \left(\frac{2\mu_1 - \lambda_1}{2\mu_1}\right) \\ &= \quad \frac{1 - r}{\mu_2} + \frac{1}{2\mu_1} - \frac{1 - r}{2\mu_1\mu_2}(\mu_1 + \mu_2). \end{aligned} \tag{4.26}$$

Figure 4.3: Type 2 customer's waiting time, $S^0 = (0, 2)$, $L^0 = (l_1, 0)$


$Exp(2\mu_2 + \lambda_1)$
$\frac{2\mu_2}{2\mu_2+\lambda_1}$
$\frac{\lambda_1}{2\mu_2+\lambda_1}$
$W_2(1, 1; l_1 - 1, 0)$
$W_2(0, 2; l_1 + 1, 0)$

iv. $\mathbf{S^0 = (0, 2)}$; both served customers are of type 2

The waiting time $W_2(0, 2; l_1, 0)$ can also be divided into two stages, as described in Figure 4.3.

The difference equation for the moment generating function $M_{W_2(0,2;l,0)}(\theta)$ is:

$$M_{W_2(0,2;l,0)}(\theta) = \frac{2\mu_2}{2\mu_2-\theta}\left(\frac{2\mu_2}{2\mu_2+\lambda_1}\cdot M_{W_2(1,1;l-1,0)}(\theta) + \frac{\lambda_1}{2\mu_2+\lambda_1}\cdot M_{W_2(0,2;l+1,0)}(\theta)\right) \quad (4.27)$$

The initial condition is immediate:

$$M_{W_2(0,2;-1,0)}(\theta) = 1. \quad (4.28)$$

$M_{W_2(1,1;l-1,0)}(\theta)$ was discussed in the previous section. If that function is known, then we are left with a first order difference equation, and a single initial condition, and the solution for the above can be found.

4.2.3 Waiting time of a low priority customer (type 2), when $\mathbf{L_2^0 > 0}$

To complete the analysis, we now review the situation when there are l_2 ($l_2 > 0$) type 2 customers already waiting, upon arrival. The discussion of the waiting time $W_2(s_1, s_2; l_1, l_2)$, is broken up to 3 parts:

i. Time until the first type 2 customer enters service.

ii. Time between service entries of first type 2 customer and of our customer.

iii. $W_2(s_1, s_2; l_1, l_2)$ - Summary.

i. Time until the first type 2 customer enters service

The first type 2 customer enters service when there are no type 1 customers waiting and one server becomes available. This period statistically equals the waiting time of a type 2 customer who finds the same service configuration, the same number of type 1 customers in queue and no type 2 customers, i.e. $W_2(s_1, s_2; l_1, 0)$. The generating function of that time was found conditioning on service configuration in Sub-section 4.2.2.

ii. Time between service entries of the first type 2 customer and of our customer:

This period consists of l_2 time periods, each represents the time between service entries of two successive type 2 customers. Each of those l_2 periods is a busy period, similarly to what was explained before.
Given the service configuration at the beginning of the busy period, the time of each busy period, (i.e. the time between successive entries of type 2 customers to service), is distributed the same as the waiting time of a type 2 customer who finds no waiting customers upon arrival ($L^0 = (0, 0)$) and the same service configuration. The moment generating function of that waiting time conditioning on the service configuration upon arrival, was found in Sub-section 4.2.2. (The functions found should be used letting $l = 0$).

Service Configuration Probabilities:

In order to compute the time of each of the l_2 busy periods, we should now examine the probabilities of the service configuration at the beginning of each period. We will define $S^{B_i} = (S_1^{B_i}, S_2^{B_i})$, ($i \geq 1$) to be the service configuration at the moment when the i'th type 2 customer enters service. We distinguish between two cases:

- Service configuration at the moment when the first type 2 customer enters service (S^{B_1}). This moment is also the end of the busy period, which started upon arrival of our customer.

- Service configuration at the end of busy period, that started upon an entry to service of a type 2 customer (S^{B_i}, $i > 1$).

The above two cases differ from each other by both service configuration and queue size at the beginning of the period. First, while the first busy period is opened with l_1^0 type 1 customers waiting in line, the following l_2^0 periods are opened when there are no type 1 waiting customers ($L_1^{B_1} = l_1$; $L_1^{B_i} = 0$, $i \geq 1$). Second, the first busy period may be opened with two type 1 customers in service, while all the following busy periods are opened with at least one type 2 customer in service. The service configuration at the beginning of the first busy period is known (S^0), unlike that of each of the next periods.
Following are explanations about the service-configuration probabilities in each of the two cases:

i. **Service-configuration probabilities just after the first type 2 customer enters service (S^{B_1}):** Those probabilities depend on the initial service configuration (S^0) and on the number of type 1 waiting customers (L_1^0).

   - $S^{B_1}|(S^0 = (2,0))$: Assuming two type 1 customers in service upon arrival, then when the first type 2 customer enters service, there will be one served customer of each type. The formal probabilities are immediate:

$$S^{B_1}|(S^0 = (2,0)) = \begin{cases} (2,0), & w.p.\ 0 \\ (1,1), & w.p.\ 1 \\ (0,2), & w.p.\ 0 \end{cases}$$

   - $S^{B_1}|(S^0 = (1,1))$: if upon arrival one customer of each type was served, then the next busy period begins (just after first type 2 customer enters service) either with one customer of each type in service (if the originally served type 2 customer has left during the busy period) or with two type 2 customers in service (if the originally served type 2 customer has not left service yet). $r = P\left(S^B = (1,1)|S^0 = (1,1)\right)$ - the probability that a type 2 customer will still be in service at the end of a busy period was already found and is given in (4.22). This probability referred to the case when at the beginning of the busy period there were one customer of each type in service, and no customer in queue. Considering that now we discuss the case when, at the beginning of the period, there were l_1 type 1 customer in queue, we obtain the following probabilities:

$$S^{B_1}|(S^0 = (1,1)) = \begin{cases} (2,0), & w.p.\ 0 \\ (1,1), & w.p.\ 1 - r^{(l_1+1)} \\ (0,2), & w.p.\ r^{(l_1+1)} \end{cases} \tag{4.29}$$

- $S^{B_1}|(S^0 = (0,2))$: if upon arrival two type 2 customers were served, then the next busy period (entry of the next type 2 customer to service) opens with a service configuration of $S^{B_i} = (0,2)$, if and only if one of the initially served type 2 customers has not left service yet. We now look at the system state at the end of the first service completion (after an $Exp(2\mu_2)$ time period).
  Let the number of type 1 customers that arrived during the period until the first service completion be denoted by A_1^0 (A_1 stands for the number of type 1 arrivals, and A_1^0 indicates the duration of the first iteration). Now we look at the probability that at the end of such a single busy period, there will still be a type 2 customer in service:

  If $A_1^0 = 0$, meaning that no type 1 customers have arrived until first service completion, then the busy period is completed. In this case the next busy period will immediately start with a service configuration of $S^{B_i} = (0,2)$. The probability for this case is $\frac{2\mu_2}{2\mu_2+\lambda_1}$.

  If $A_1^0 > 0$, then the probability that at the end of the busy period, there will still be a type 2 customer in service is $r^{A_1^0}$, when $A_1^0 \overset{d}{=} G(\frac{2\mu_2}{2\mu_2+\lambda_1})$. (The number of type 1 customers that arrive during the $Exp(2\mu_2)$ period has a Geometric distribution with probability $\frac{2\mu_2}{2\mu_2+\lambda_1}$).

  Let $\grave{\mathbf{r}}$ be the probability that a type 2 customer will still be served at the end of a busy period, started with two type 2 served customers:

$$\grave{r} = P(S^{B_i} = (0,2)|S^{B_{i-1}} = (0,2)). \tag{4.30}$$

  Based on the above:

$$\begin{aligned} \grave{r} &= \sum_{a=0}^{\infty} r^a \cdot (\frac{\lambda_1}{2\mu_2+\lambda_1})^a \cdot \frac{2\mu_2}{2\mu_2+\lambda_1} \\ &= \frac{2\mu_2}{2\mu_2+\lambda_1(1-r)} \end{aligned} \tag{4.31}$$

  (Again, for definition of r see equation (4.21)and for solution of r (4.22)). The above calculation applies to a busy period started with no customers in queue. For a type 2 customer to still be in service after all l_1

busy periods (opened by each of the type 1 customers originally waiting), we need that a type 2 customer will still be in service at the end of the first busy period, which is opened with $S = (0, 2)$, and then that he will still be served during l_1 busy periods, each opened with $S = (1, 1)$. The service configuration probabilities are therefore:

$$S^{B_1}|(S^0 = (0,2)) = \begin{cases} (2,0), & w.p. \quad 0 \\ (1,1), & w.p. \quad 1 - \grave{r} \cdot r^{l_1} \\ (0,2), & w.p. \quad \grave{r} \cdot r^{l_1} \end{cases} \tag{4.32}$$

ii. **Service configuration probabilities for (S^{B_i}, $i > 1$):** As mentioned before, at the beginning of each period there is at least one type 2 customer in service. The service configuration is therefore either $S^{B_i} = (1, 1)$ or $S^{B_i} = (0, 2)$.

S^{B_i} is a Markov chain, that receive the values (1,1) or (0,2) for every $i = 1, .., l_2$ and the values (2,0), (1,1), (0,2) for $i = 0$. The fact that $S^{(i)}$ is a Markov chain, is now proven and will be used also later in this work.

Proof:

$$\begin{aligned} & P\{S^{(B_i)} = (0,2)|S^{B_{(i-1)}} = (1,1), S^{B(i-2)}, ..., S^{B_{(0)}}\} \\ = \quad & Eq^{(A_1^{B_i}+1)}|S^{B_{(i-1)}} = (1,1), .., S^{B_{(0)}} \\ = \quad & Eq^{(A_1(1,1)+1)} \\ = \quad & P\{S^{B_{(i)}} = (0,2)|S^{B_{(i-1)}} = (1,1)\}. \end{aligned}$$

Similarly:

$$\begin{aligned} & P\{S^{B_i} = (0,2)|S^{B_{i-1}} = (0,2), S^{B_{i-2}}, ..., S^{B_0}\} \\ = \quad & Eq^{A_1^{B_i}}|S^{B_{i-1}} = (0,2), ..., S^{B_0} \quad = \quad Eq^{A_1(0,2)} \\ = \quad & P\{S^{B_i} = (0,2)|S^{B_{i-1}} = (0,2)\} \end{aligned}$$

The probability matrix of S^{B_i}, $i \geq 1$ is :

$$\begin{array}{c|ll} & (1,1) & (0,2) \\ \hline (1,1) & 1 - Eq^{(A_1(1,1)+1)} & Eq^{(A_1(1,1)+1)} \\ (0,2) & 1 - Eq^{A_1(0,2)} & Eq^{A_1(0,2)} \end{array}$$

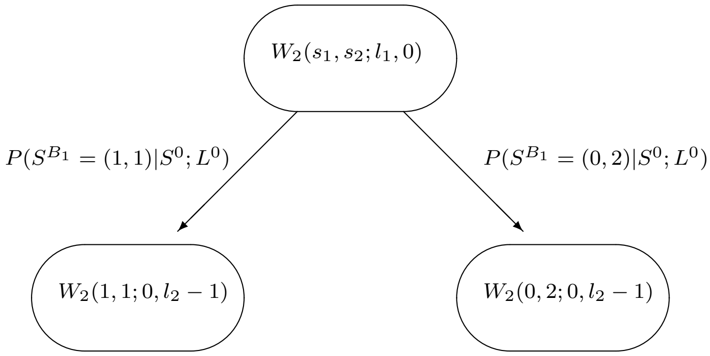
Using the relations $Eq^{(A_1(1,1)+1)} = r$ and $Eq^{A_1(0,2)} = \grave{r}$, the probability matrix can also be written as:

$$\begin{array}{c|cc} & (1,1) & (0,2) \\ \hline & & \\ (1,1) & 1-r & r \\ (0,2) & 1-\grave{r} & \grave{r} \end{array}.$$

iii. $\mathbf{W_2(s_1, s_2; l_1, l_2)}$ - Summary

All the previously discussed components are combined to produce the total waiting time $W_2(s_1, s_2; l_1, l_2)$. That is also demonstrated in Figure 4.4.

Figure 4.4: Total waiting time of a low priority customer



The total waiting time can be represented by the following two stage combination:

$$W_2(s_1, s_2; l_1, l_2) \overset{d}{=} W_2(s_1, s_2; l_1, 0) + \begin{cases} W_2(1, 1; 0, l_2 - 1) & w.p \quad P(S^{B_1} = (1,1)|S^0; L^0) \\ \\ W_2(0, 2; 0, l_2 - 1) & w.p \quad P(S^{B_1} = (0,2)|S^0; L^0) \end{cases},$$

where

- $W_2(s_1, s_2; l_1, 0)$ was discussed and found in the Subsection 4.2.2.

- $P(S^{B_1} = (1,1)|S^0; L^0)$ and $P(S^{B_1} = (0,2)|S^0; L^0)$ were explicitly found and can be denoted by r in(4.22)and $\grave{r}$ in (4.31).

- $W_2(s_1, s_2; 0, l_2 - 1)$ can be recursively computed.

Chapter 5

Exact Analysis: Matrix Geometric Solutions

5.1 The matrix geometric method

The theory of matrix geometric solutions was pioneered by Marcel Neuts (see [23]). The technique generalizes solutions of scalar-matrices, allowing to replace scalars with blocks. The technique can be applied to Markov processes with a repetitive form, in order to calculate stationary probabilities and time to absorption. A Markov process has a repetitive state form, if the transition rate from state (i, j) to state $(i+k, \grave{j})$ is independent of the value of i for some $\grave{i}$ $(i > \grave{i})$. In terms of the generator matrix, the repetitive form implies that matrix entries eventually repeat diagonally.

Specific examples of Markov processes with a repetitive form are birth and death processes, and quasi birth and death processes. While a birth and death process allows only adjacent state transitions, in a quasi birth and death process such adjacent neighbor transitions are interpreted in terms of vectors of states. The result is a transition rate matrix with a block-tridiagonal structure rather than a scalar-tridiagonal structure in the birth-and-death case. The matrix geometric method, allows to implement the results obtained for scalar matrices in birth and death processes, to block-matrices in quasi birth and death processes.

The matrix geometric technique can also be applied to find the distribution of a continuous phase-type variable. A continuous phase-type distribution is defined as the distribution of time until absorption in an absorbing Markov process.

The waiting time of a customer in any service system with Markovian characteristics (all times are exponential, and the appropriate independence assumptions) can be denoted by a phase-type distribution. System state is presented by an appropriate vector. Absorption occurs when the customer of interest enters service.

All other system states (describing cases in which our customer is still in line) are considered transient. In [22] (Section 9.5), it is demonstrated how to use the technique in order to find distributions of time to absorptions and of steady-state for a Markov process with an infinite number of states. In this work, we focus on calculating waiting times in systems where queue size is unlimited. The matrix geometric solution can only be applied to systems in which one element at most can get infinite number of values. As will be explained later on, that prevents us from being able to use the method for a general case of more than two service types, where the queue size is unlimited. In practice, however, queue size is limited, even if by a very large number. Therefore, it may be actually more practical to assume a finite system. However, in our 'simple case' example we stay with the infinite system, as originally defined.

In this work, we introduce the option of matrix geometric technique by presenting some specific results and demonstrating how to use them for estimation of waiting times. To find more about the theory and its development, the reader is referred to [23].

First, we shall present the general result in [22]. Then we demonstrate how to use that result in order to calculate waiting times in the simple case of two servers, two service types, and priorities, (the model analyzed in Chapter 4). As will be demonstrated, this involves the inversion of infinite matrices. To summarize the chapter, we will explain in broad terms how to use the technique for additional models.

5.2 The distribution of a continuous phase-type distribution

To calculate an expression for the distribution function of a phase- type distribution, we let the generator matrix be partitioned as follows:

$$\left[\begin{array}{c|c} U & A \\ \hline 0 & 0 \end{array}\right], \tag{5.1}$$

when U is an (nXn) matrix defining the transition rates between the transient states $\{1, 2, ..., n\}$, and A is a $(nX1)$ vector that defines the transition rates between the transient states and the absorbing state $n + 1$. If the time to absorption is denoted by W, then

$$E[W^n] = (-1)^n n! \beta U^{-n} e \tag{5.2}$$

and

$$E[W] = -\beta U^{-1} e, \tag{5.3}$$

when β is the initial distribution. It is also possible to find $P(t)$, the probability that the process has not reached absorption by time t

$$P(t) \;=\; \beta \cdot e^{Ut}. \tag{5.4}$$

5.3 An example: calculating waiting times in the 'simple case'

Recall the 'simple case' described in Chapter 4. In that model there are two statistically identical servers and two service types, when type 1 has a non-preemptive priority over type 2. We now demonstrate how the moments of the waiting time of a type 2 customer can be calculated using the matrix geometric technique. We will begin with the case of no type 2 customers waiting ($L_2 = 0$) upon arrival, and then proceed to the case of any number of waiting customers.

5.3.1 Waiting time of a type 2 customer, $L_2 = 0$

First, we shall describe the process as a Markov jumping-process. We define a two dimensional state (i,j) where $i \in \{0, 1, 2\}$ stands for the number of type 1 customers in service, and $j \in \{-1, 0, 1, ...\}$ is the number of type 1 waiting customers. $j = -1$ describes the state of no waiting customers and one available server, meaning that our customer enters service. ($(i, j) = (1, -1)$ if there is a type 1 customer in service and our customer is just about to enter service, and $(i, j) = (0, -1)$ if the customer served is a type 2 customer). Note, that this description of system state, is shorter than the one used in Chapter 4. Figure 5.1 describes the states diagram.

The transition rates in Figure 5.1 are translated into the generator matrix Q

$$Q \;=\; \begin{array}{c|cc|ccc|ccc|ccc|c}
 & (0,-1) & (1,-1) & (0,0) & (1,0) & (2,0) & (0,1) & (1,1) & (2,1) & (0,2) & (1,2) & (2,2) & \dots \\
\hline
(0,-1) & & & & & & & & & & & & \\
(1,-1) & & & & & & & & & & & & \\
\hline
(0,0) & 2\mu_2 & 0 & \alpha_0 & 0 & 0 & \lambda_1 & 0 & 0 & & & & \\
(1,0) & \mu_1 & \mu_2 & 0 & \alpha_1 & 0 & 0 & \lambda_1 & 0 & & & & \\
(2,0) & 0 & 2\mu_2 & 0 & 0 & \alpha_2 & 0 & 0 & \lambda_1 & & & & \\
\hline
(0,1) & & & 0 & 2\mu_2 & 0 & \alpha_0 & 0 & 0 & \lambda_1 & 0 & 0 & \\
(1,1) & & & 0 & \mu_1 & \mu_2 & 0 & \alpha_1 & 0 & 0 & \lambda_1 & 0 & \\
(2,1) & & & 0 & 0 & 2\mu_1 & 0 & 0 & \alpha_2 & 0 & 0 & \lambda_1 & \\
\hline
(0,2) & & & & & & 0 & 2\mu_2 & 0 & \alpha_0 & 0 & 0 & \\
(1,2) & & & & & & 0 & \mu_1 & \mu_2 & 0 & \alpha_1 & 0 & \dots \\
(2,2) & & & & & & 0 & 0 & 2\mu_1 & 0 & 0 & \alpha_2 & \\
\hline
 & & & & & & & & & & & &
\end{array}$$

when

$$\alpha_0 \;=\; -(\lambda_1 + 2\mu_2) \tag{5.5}$$

$$\alpha_1 \;=\; -(\lambda_1 + \mu_1 + \mu_2) \tag{5.6}$$

$$\alpha_2 \;=\; -(\lambda_1 + 2\mu_1) \tag{5.7}$$

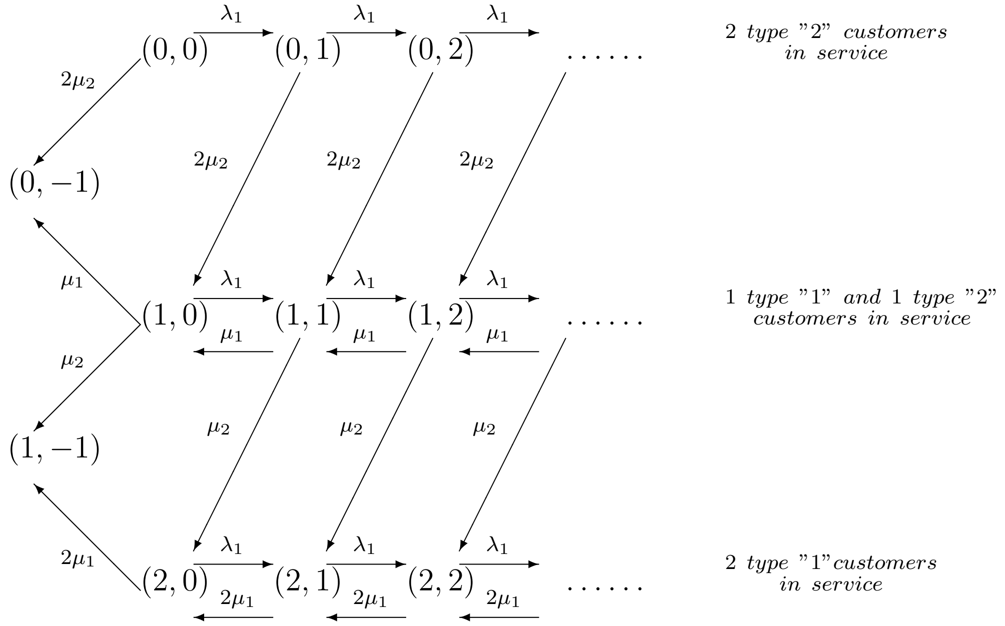
We are interested in the time to absorption of that process (the time until the process reaches state $(0,-1)$ or $(1,-1)$). We group the two states into one absorbing state and rewrite the generator matrix Q in the form:

$$Q \;=\; \left[\begin{array}{c|c} U & A \\ \hline 0 & 0 \end{array}\right]. \tag{5.8}$$

U represents the transition rates between the transient states, and A is a column that represent the transition rates from the transient states to the absorbing one. Being more precise, the matrices U and A are denoted by:

$$U \;=\; \left(\begin{array}{c|ccc|ccc|ccc|c}
 & (0,0) & (1,0) & (2,0) & (0,1) & (1,1) & (2,1) & (0,2) & (1,2) & (2,2) & \ldots \\ \hline
(0,0) & \alpha_0 & 0 & 0 & \lambda_1 & 0 & 0 & & & & \\
(1,0) & 0 & \alpha_1 & 0 & 0 & \lambda_1 & 0 & & & & \\
(2,0) & 0 & 0 & \alpha_2 & 0 & 0 & \lambda_1 & & & & \\ \hline
(0,1) & 0 & 2\mu_2 & 0 & \alpha_0 & 0 & 0 & \lambda_1 & 0 & 0 & \\
(1,1) & 0 & \mu_1 & \mu_2 & 0 & \alpha_1 & 0 & 0 & \lambda_1 & 0 & \\
(2,1) & 0 & 0 & 2\mu_1 & 0 & 0 & \alpha_2 & 0 & 0 & \lambda_1 & \\ \hline
(0,2) & & & & 0 & 2\mu_2 & 0 & \alpha_0 & 0 & 0 & \\
(1,2) & & & & 0 & \mu_1 & \mu_2 & 0 & \alpha_1 & 0 & \ldots \\
(2,2) & & & & 0 & 0 & 2\mu_1 & 0 & 0 & \alpha_2 & \\ \hline
 & & & & & & & & & &
\end{array}\right) \tag{5.9}$$

Figure 5.1: Waiting time of a type 2 customer, $L_2 = 0$ - a Markov jumping process



and

$$A = \begin{pmatrix} 2\mu_2 \\ \mu_1+\mu_2 \\ 2\mu_1 \\ 0 \\ 0 \\ \vdots \end{pmatrix} \tag{5.10}$$

Recall that if W is the time to absorption, then according to (5.2) and (5.3)

$$\begin{aligned} E\left[W\right] &= -\beta U^{-1}e. \\ &\textit{and} \\ E\left[W^n\right] &= (-1)^n n!\beta U^{-n}e, \end{aligned}$$

when β is the initial distribution. In our case β is well known, since the initial state is known.

In order to obtain an explicit expression for the moments of W (or for its distribution function), one should invert the infinite matrix U. Doing this for some general μ_1, μ_2, λ_1 is not immediate, and we suspect that an explicit expression can not be obtained. It is possible, however, to limit queue size and use a finite matrix U.

5.3.2 Waiting time of a type 2 customer, $L_2 > 0$

When type 2 customers were already waiting upon arrival of our customer, the system state is described by a 3 dimensional vector (i,j,k), where i and j are the same as in 5.3.1, and k is the number of type 2 customers waiting before our customer. Absorption states are $(i,-1,0)$ for $i \in \{0,1\}$. Note that j is the only element that can take infinite values.

Transition rates are determined as follows:

- For states within the same k (same number of type 2 customers in line), transition rates are the same as in 5.3.1.

- Transition rate from state $(i,0,k)$ to $(i-1,0,k-1)$ is $\mu_1 \cdot i$ $(i>0)$.

- Transition rate from state $(i,0,k)$ to $(i,0,k-1)$ is $\mu_2 \cdot (2-i)$.

We now explain the structure of the matrix U, that denotes the transition rates between transient states. For convenience, we order the states in blocks, so that all the states in the same block have the same value of j (type 1 waiting customers). Each such block is divided into sub-blocks, so that all the states in a sub-block have also the same value of k (number of type 2 waiting customers). In a single sub-block there are 3 states, each representing a possible value of i (number of type 1 customers in service). Transitions between states with different values of k are available only when there are no type 1 waiting customers, meaning when $j = 0$. That implies that transitions between states with different k are allowed only for states in the first block of the matrix. This part of the matrix is structured as follows:

$$
\begin{array}{c||c ccc|ccc|c|ccc|c}
 & \dots & (0,0,k-1) & (1,0,k-1) & (2,0,k-1) & (0,0,k) & (1,0,k) & (2,0,k) & \dots & (0,1,k) & (1,0,k) & (2,0,k) & \dots \\
\hline\hline
\vdots & & & & & & & & & & & & \\
\hline
(0,0,k-1) & & \alpha_0 & 0 & 0 & & & & & & & & \\
(1,0,k-1) & & 0 & \alpha_1 & 0 & & & & & & & & \\
(2,0,k-1) & & 0 & 0 & \alpha_2 & & & & & & & & \\
\hline
(0,0,k) & & 2\mu_2 & 0 & 0 & \alpha_0 & 0 & 0 & & \lambda_1 & 0 & 0 & \\
(1,0,k) & & \mu_1 & \mu_2 & 0 & 0 & \alpha_1 & 0 & & 0 & \lambda_1 & 0 & \\
(2,,0,k) & & 0 & 2\mu_1 & 0 & 0 & 0 & \alpha_2 & & 0 & 0 & \lambda_1 & \\
\hline
(0,0,k+1) & & & & & 2\mu_2 & 0 & 0 & & & & & \\
(1,0,k+1) & & & & & \mu_1 & \mu_2 & 0 & & & \dots & & \\
(2,0,k+1) & & & & & 0 & 2\mu_1 & 0 & & & & & \\
\hline
\vdots & & & & & & & & & & & & \\
\hline\hline
(0,1,0) & & & & & & & & & & & & \\
(1,1,0) & & & & & & & & & & & & \\
(2,1,0) & & & & & & & & & & & & \\
(0,1,1) & & & & & & & & & & & &
\end{array}
$$

When $j > 0$, transitions are available only within the same k, and transition rates are, therefore, the same as in 5.3.1.

The vector A of transition rates from transient states to the absorbing one is:

$$
A \;=\; \begin{pmatrix} 2\mu_2 \\ \mu_1 + \mu_2 \\ 2\mu_1 \\ 0 \\ 0 \\ \vdots \end{pmatrix}.
$$

As already mentioned in 5.3.1, calculations involve the inversion of an infinite matrix U. We do not have the result of that inversion, and we suspect that a general symbolic result can not be obtained.

5.4 Calculating waiting times for more general cases

Though we did not reach a symbolic result for U^{-1} or an explicit expression for the distribution of waiting time, such results may be numerically obtained. We therefore believe that using the described method can be of value, and will now discuss in broad terms, the issue of implementing it for more general models.

Only few changes to the representation of the system state and to the matrices U and A should be applied, when the number of servers or the number of service types increases.

When the number of servers is s, the system state is represented by a three dimensional vector, similarly to that in Sub-section 5.3.2. The first element i, which stands for the number of type 1 served customers receives values in $\{0, 1, ..., s\}$. The absorption states are now the $s+1$ states of the form $i, -1, 0$, where $i \in \{0, 1, ..., s\}$. Transition rates are defined by the system service rate at any given state. For example from the state (i, j, k) $(j > 0,\, k > 0)$, it is possible to move to $(i+1, j-1, k)$ with a rate of $\mu_2 \cdot (s-i)$, to state $(i, j-1, k)$ with a rate of $\mu_1 \cdot i$ or to state $(i, j+1, k)$ with a rate λ_1.

If k different service types are provided by the call center (k may be larger than two), the system state should be represented by a vector of the appropriate dimension to include information regarding how many customers of each type are being served and how many customers of each type are waiting (the minimal dimension to describe this is $2k-1$). As already mentioned, the method of matrix geometric is applicable only when at most one element of the state-vector can take on an infinite number of values. In the case described, there are $k-1$ components that can take on infinite values (components that stand for the number of waiting customers of types $\{1, 2, ..., k-1\}$). This implies that we might not be able to use the matrix geometric technique to estimate waiting times for this model. However, we can do so, if we are interested in the waiting time of a type 1 or type 2 (out of k types) customer (and then we are not interested in future arrivals of types greater than 1), or when service rates of the different types are the same (in this case, the system state can be represented by a 2 dimensional vector). In other cases, it is possible to limit the queue size (even if by a large number), and to solve for a finite system.

Chapter 6

Inexplicit Analytical Approximation

As was demonstrated by previous analysis, it might be difficult to explicitly estimate the waiting time distribution even for relatively simple models, when priority discipline is implemented.
Usually, real life systems are more challenging than the models discussed, as additional parameters get into the picture:

- A large number of servers
- Different servers have different skills
- Different priorities implemented by different servers (according to skills)
- Abandonment
- More than two service types

Also, since we are dealing with a "real time" problem, computation should be immediate, otherwise it becomes useless. Other than the exact calculation, we have two options for estimating the waiting time:

i. Simulation

ii. Approximation

Since simulations are usually time consuming, we would like to suggest an alternative.

In this chapter we propose a method for approximating waiting times. We present in general lines the method and the rational behind it. In the next chapters we discuss accuracy levels of parts of the approximation for the 'simple case' defined previously in this work. In Chapter 8, we present a detailed algorithm.

6.1 The method

The suggested approximation had been first developed by Prof. Isaac Meilijson, from the School of Mathematics in the Faculty of Exact Sciences, Tel-Aviv University. We elaborated on the method, to support more general cases.

We approximate the waiting time by imitating a real-life system, while replacing complicated distributions with deterministic values or with simple distributions. The method is briefly described by the following principles:

- The algorithm is iterative. Each iteration corresponds to the time between successive completions of service.
- The mean time of each iteration is calculated and is added to the total waiting time, weighted by the probability that the customer in interest is still waiting at the beginning of the iteration.
- The mean iteration time is approximated based on service configuration probabilities.
- At the end of each iteration, service configuration probabilities for the next iteration are updated. The method is designed for systems, in which servers are not necessarily statistically identical. Therefore, information regarding how many customers of each type are being served is not enough for calculating iteration time, and information regarding who are they being served by is essential. Service configuration probabilities are being re-calculated in every iteration, in two steps:
    - The probability of each server to be the one who completes service in the next service completion event is calculated.
    - New tasks (from the queue) are assigned to servers, according to these probabilities.
- The number of arrivals in each iteration is approximated by its average.
- New arrivals join the queue, and so affect the number of iterations. The number of iterations is approximated by the initial queue size and the approximated number of arrivals.
- The probability of each customer to enter service (as opposed to be still waiting) is also updated at the end of each iteration.

6.2 Marking convention

Before getting into details, we define the following notations:

- $\mathbf{Sv}$ is the number of servers in the system.

- $\mathbf{Tp}$ is the number of service types.

- $\mathbf{n_0}$ is the index (initial place in the system) of the customer in interest (the customer for whom we estimate the waiting time).

- $\mathbf{tot^i(k)}$ is the probability that k'th customer has entered service before the i'th iteration. The probability that the k'th customer is still waiting during the i'th iteration is respectively denoted by $(1 - tot^i(k))$.

- $\bar{\mu}^{\mathbf{i}}$ is the accumulated service rate in iteration i.

- $\mathbf{P^i_{[SvXTp]}}$ is the service configuration probabilities matrix in the i'th iteration. $\mathbf{P^i(k, j)}$ is the probability that server k serves a type j customer during iteration i.

- $\mathbf{T^i}$ is the duration between the $(i-1)'th$ and the $i'th$ departures, or, in other words, the duration of the i'th iteration .

- $\mathbf{A^i_1}$ is the number of type 1 customers that arrive during the i'th iteration.

- $\mathbf{N}$ is the number of iterations that our customer waits.

- $\mathbf{X^i_{[Sv]}}$ is a vector of the dimension of number of servers, denoting the exact service configuration at a given moment. $X^i(k)$ is the type of customer served by server k in the i'th iteration.

- $\mathbf{S^i_{[Tp]}}$ denotes the number of customers of each type that are being served in the i'th iteration.

- $\mathbf{L^i_{[Tp]}}$ as before is the queue size for each type of customers in the i'th iteration.

- $\mathbf{W_j(S^0; L^0)}$ or $\mathbf{W_j(X^0; L^0)}$ is the waiting time of the n_0'th customer, given that he is a type j customer and given the system state at the time of his arrival $((S^0; L^0)$ or $(X^0; L^0))$.

Sometimes, when we speak about any single iteration and not about a specific one i we omit time indication. For example, we may use A_1 to specify the number of arrivals in a single iteration, when it is not important to which specific iteration we refer, or when we refer to the current iteration.

6.3 The rational behind the method

We shall refer to several issues:

i. Calculating the waiting time as the sum of iteration times.

ii. Calculating the time of each iteration, given service configuration probabilities.

iii. Calculating service configuration probabilities.

iv. Approximating the number of iterations.

6.3.1 Calculating the waiting time as the sum of iteration times

The expected waiting time is calculated as the sum of small parts of the waiting time. Each part is the time between successive completions of service. Each iteration is added to the expected waiting time, weighted by the probability that the customer under interest has not entered service yet. Formally:

$$E[W_k(S;L)] = E[\sum_{i=1}^{N} T^i],$$

which may be developed to:

$$E[W_k(S;L)] = E[\textstyle\sum_{i=1}^{N} T^i] = E\sum_{i=1}^{\infty} 1_{i\le N} T^i.$$

Since the event $N < i$ is independent of the duration of i'th iteration T^i, or, formally:

$$(N < i) \perp T^i,$$

then

$$\begin{aligned} E[W_k(S;L)] &= \sum_{i=1}^{\infty} E[1_{i\le N} \cdot T^i] \\ &= \sum_{i=1}^{\infty} E(T^i) \cdot P(^{customer\ is\ still\ waiting}_{\ during\ i'th\ iteration}) \\ &= \sum_{\mathbf{i;tot(n_o)^i<1}} [\frac{1}{\mu^i} \cdot (1 - tot^i(n_0))]. \end{aligned}$$

6.3.2 Calculating the time of each iteration

Given that exact service configuration, X, iteration time is exponentially distributed, and average iteration time is $\frac{1}{\sum_{k=1}^{Tp}\mu_{X(k)}}$. However, we only have the probabilities for each server to be serving each type of customer, and we do not have the exact service configuration X. Therefore, average iteration time should be calculated based on the law of total probability, going over all possible values of X. That might be very time consuming for large systems.

We, therefore, suggest to use an approximation. The service time of a specific server is an hyper-exponential variable. Its mean can be calculated based on the probability vector and service rates, and is given by: $\sum_{j=1}^{Tp}\frac{P(i,j)}{\mu_j}$. Iteration time is denoted by the minimum of all servers' service time, and, hence, it is a minimum of hyper-exponentials. We approximate each hyper-exponential (for each server's service time) by an exponential variable. Then we approximate iteration time by the minimum of these exponentials. As a result, iteration time is approximated by an exponential with a rate $\bar{\mu}$, where $\bar{\mu}$ is the sum of all servers' approximated rates. Mean iteration time is, therefore, approximated by $\frac{1}{\bar{\mu}}$.

Intuitively, approximating the hyper-exponential service time (of a specific server) by an exponential variable is expected to work better when service configuration (for that server) is known with a high probability, or when service rates of different types are similar. Therefore, we expect the approximation of iteration time to work better when service configuration of all servers is known with a high probability or when service types do not significantly differ by the service times required.

However, when the number of servers is small, and assuming that the service types served by different customers are independent (which is not always correct), we can calculate the probability of each value of X. (We have implemented a recursive procedure for such a computation. The procedure is given in Appendix A).

6.3.3 Calculating service configuration probabilities

In the initial stage (first iteration) the service configuration is known. The matrix is, therefore, a binary matrix. Then, the matrix is updated at the end of each iteration, in two steps:

i. Calculating the probability for each server to be the one who completes service at the end of iteration.

ii. Assigning tasks (or jobs) to servers for the next iteration.

We will now discuss each of the steps.

i. Probabilities of service completion

We are now interested in the probability that server k will be the one who completes service in the i'th iteration. We let the approximation of this probability be denoted by $Pes^i(k)$. In terms of fluid approximations, $Pes^i(k)$ is the fraction of the work that server k is expected to finish during the iteration.
Recall that the system service configuration is denoted by $\mathbf{X}$ (receiving values $x \in \Omega_X$). Given that server k is serving a type j customer, and given service probabilities for the other servers, the accurate probability that server k will be the first to complete service is:

$$\sum_{x\in\Omega_x;x(k)=j} \frac{\mu_j}{\sum_{l=1}^{Sv} \mu_{x(l)}} \cdot P(X=x|X_k=j) = \sum_{x\in\Omega_x;x(k)=j} \frac{\mu_j}{\sum_{l=1}^{Sv} \mu_{x(l)}} \cdot \pi_{l=1;l\neq k}^{Sv} P^i(l,x(l)).$$

However, going over all service configurations is very time consuming when we deal with large systems. Therefore, here again, we use the exponential approximation for the service times. We assume that server k is serving a type j (with probability P(k,j)). Given that, his service time is exponential with a rate μ_j. We assume that all other servers' service times are also exponential. The probability that server k will be the first to complete service, given that he is serving a type j customer can now be approximated as the minimum of exponentials:

$$Pes^i(k)|(x(k)=j) = \frac{\mu_j}{\bar{\mu}^i - \mu(\bar{k}) + \mu_j},$$

when $\mu(\bar{k}) = \frac{1}{\sum_{l=1}^{Ty} \frac{P(k,l)}{\mu_l}}$ and is the effective unconditional service rate of server k.

This part of the approximation is similar to the approximation of iteration times. Therefore, the conditions under which it is a good approximation are similar to the conditions discussed in the previous sub-subsection.

ii. Task assignment

Once we have the probability for each of the servers to be "available" at the end of iteration, we assign tasks for the next iteration. We look for the next customer (the longest-waiting customer among the customers with highest priority). The most accurate way to assign jobs is to split the work of a single customer between the servers proportionally to their service completion probabilities. The problem with that method, is, again, that it becomes inefficient (in terms of calculation time) as the number of servers increases. We therefore arbitrarily assign the maximum amount possible from the next job to the first available server and only then move to the next server. We believe, that the inaccuracy of doing so decreases as the system load increases.

Another way to look at the assignment algorithm is as a fluid approximation. The work required by each customer is divided into small fractions. The work is not necessarily being cared of at once. The probability that a server is available may be thought of as his capacity. It is possible to take care of fractions of the work in different iterations or at different servers, according to servers capacity and to fraction of work of each customer which are left. For example: if the probability that a specific server i will be the one to complete service at the end of the iteration is 0.75, then we will assign that server 0.75 of the next job. If the probability of this next job to be still waiting is lower than 0.75, than we will assign the remaining job (update its probability to be waiting to 0), and assign additional fractions of work from the next customer. In every iteration a server dedicates part of the time to different customers (according to service configuration probabilities). A customer is considered as leaving the queue in order to be served, exactly after all the fractions of the work he requires, entered service. As long as there is any amount of work waiting, the customer is considered waiting.

6.3.4 Number of iterations

The number of iterations might be critical for estimating the waiting time. It is determined by the queue size at the time of arrival (given) and the number of customers that arrive during our customer waiting time, and pass him in line.

We replace the stochastic number of arrivals in each iteration with its average - a deterministic value. Following is a short discussion on when such an approximation may be justified.

Recall that $A_j^i \overset{d}{=} Pois(\lambda_j \cdot T^i)$, where T^i stands for the duration between the $(i-1)$'th and i'th departures from service. The average number of customers that arrive during iteration i is, therefore:

$$E[A_j^i] \;=\; \lambda_j \cdot E[T^i]$$

and its variance:

$$\begin{aligned} Var[A_j^i)] &= E[(A_j^i)^2] - E^2[A_j^i] \\ &= \lambda_j \cdot E[T^i] + \lambda_j^2 \cdot Var[T^i] \end{aligned}$$

- According to Chebychev's inequality:

  $$P(|A_j^i - E[A_j^i]| > \epsilon) \;\leq\; \frac{Var[A_j^i]}{\epsilon^2}$$

  The above inequality implies that when the average and variance of iteration time are small, the upper bound for the probability that the real number of arrivals is far from its average, decreases. In general, the mean time of iteration decreases as the service rate of the system increases (high service rates or a large number of servers). The variance decreases when different service times of different service types are identically distributed, and when the service configuration is known with a high probability.

- Intuitively, we may replace a random variable with its average when its standard deviation is small relatively to its average value. Given the service configuration, the number of type j arrivals during an iteration time is geometrically distributed. If the service rate is given by $\overline{\mu}$, then the success probability of that geometric variable is $\frac{\overline{\mu}}{\overline{\mu}+\lambda_j}$ and the average number of arrivals is $\frac{\lambda_j}{\overline{\mu}}$. The variance of that variable is $[(\overline{\mu}+\lambda_j)\cdot\frac{\lambda_j}{\overline{\mu}^2}]$.
  The ratio between the average number of arrivals and its standard deviation is therefore:

  $$\frac{E^2(A_j^i)}{Var(A_j^i)} = \frac{\lambda_j}{\overline{\mu}+\lambda_j} = \frac{1}{1+\frac{\overline{\mu}}{\lambda_j}}$$

  We may conclude that replacing the number of arrivals in each iteration by its average has a smaller effect on the results as $\overline{\mu}$ increases or as λ_j decreases. It should be mentioned, that here, again, we assumed that iteration times are exponential. (This assumption led us to the conclusion that number of arrivals is a geometric variable).
  Generally speaking, the number of arrivals during the i'th iteration is a Poisson variable with a rate of $(\lambda_j \cdot T^i)$. The ratio between its average and standard deviation is given by:

  $$\frac{E^2(A_j^i)}{Var(A_j^i)} = \frac{1}{\frac{1}{\lambda_j E(T^i)} + \frac{Var(T^i)}{E^2(T^i)}}$$

Above we have tried to identify under what circumstances the number of arrivals in each iteration can be replaced with its average. However, less is actually needed. Although we approximate the number of arrivals separately for each iteration, the approximation accuracy is actually affected by the total number of arrivals (summing up over all iterations). Assume that the number of iterations is N and that iteration times T^i $(i = 1, .., N)$ are identically distributed. In this case we may use the weak law of large numbers to conclude:

$$\lim_{N\to\infty} P\left(|\bar{A}_j^i - E(A_j^i)| < \epsilon\right) = 1,$$

where

$$\bar{A}_j^i \quad = \quad \frac{\sum_{i=1}^{N} A_j^i}{N}.$$

This implies that:

$$\lim_{N\to\infty} \frac{\sum_{i=1}^{N} A_j^i}{N} = E(A_j^i)$$

Apparently, this justifies replacing the number of arrivals by its average when we are handling a large number of iterations (eg when the queue size is big). However, the number of iterations, N, depends on the random variables A_j^i. A_j^i $(i = 1, 2, ..., N)$ are not necessarily identically distributed. Therefore, the above justification holds only in some cases. For example, in the 'simple case' of two servers and two service types, it holds when there are no type 2 customers in queue and only type 1 customers are served upon arrival.

In the next chapter we evaluate the accuracy of parts of the approximation for the 'simple case'. We demonstrate that the waiting time calculated is not a linear function of number of arrivals A_1, but is a function of some q^{A_1}. We explain why we believe that approximating the number of arrivals might be a significant source of inaccuracies in our approximation. We also discuss the conditions, under which we believe the approximation yields a high accuracy level.

Chapter 7

Estimating the Accuracy of Our Approximation

In this chapter we review parts of the approximation in order to understand when high accuracy can be expected. Especially, we are interested in cases where queues are long. All the analysis in this chapter is done for the 'simple case', which is defined and analyzed in Chapter 4 (two statistically identical servers; two service types: type 1 has a non preemptive priority over type 2; no abandonment). We believe that the analysis can be implied to systems with more than two servers, and partially even to general systems. In Chapter 9 we go back to this subject and evaluate the accuracy of our approximation for different systems by numerically comparing its results to simulation results.

Specifically, we focus on evaluating the accuracy of replacing the number of arrivals with its average and of the exponential approximation of iteration times. We do not refer to the job assignment mechanism, which is left for future works. (Note, that in the relatively simple case of two servers and two service types, we can use improved procedures to calculate both average iteration times and job assignment probabilities, and see Appendix A for the procedure). For simplicity, each of the approximation modules is evaluated separately.

Approximation of iteration times can be similarly evaluated through the waiting time of either type 1 or type 2 customers. Approximation of number of iterations is only relevant for type 2 customers. We, therefore, selected to present the evaluation as follows:

i. Accuracy of waiting time approximation for high priority customers (type 1) - Iteration times

ii. Accuracy of waiting time approximation for low priority customers (type 2) - Number of iterations

When we discuss the waiting time of high priority customers we only refer to the approximation of iteration times. When we discuss the waiting time of low priority customers we refer only to the number of iterations.

For each of the above cases, we discuss the accuracy of the approximation separately for 3 different service configurations:

i. $\mathbf{S^0} = (\mathbf{2}, \mathbf{0})$; Two type 1 customers are being served upon arrival

ii. $\mathbf{S^0} = (\mathbf{1}, \mathbf{1})$; One type 1 customer and one type 2 customer are being served upon arrival

iii. $\mathbf{S^0} = (\mathbf{0}, \mathbf{2})$; Two type 2 customers are being served upon arrival

This is similar to the way we analyzed the system using different equations (Chapter 4). However, the results and expressions found in that chapter are not being used here, and the waiting time is described in an alternate way as the sum of all iterations.

Parameters and notations used in this chapter are as was defined in Section 6.2. We denote approximated values by $\widetilde{value}$.

7.1 Estimating the waiting times of high priority customers - iteration times

Recall that T^i is the time of the i'th iteration. $W_1(s_1, s_2; l_1, \cdot)$, the waiting time of a type 1 customer is:

$$W_1(s_1, s_2; l_1, \cdot) = \sum_{i=0}^{l_1} (T^i) \tag{7.1}$$

$$E[W_1(s_1, s_2; l_1, \cdot)] = \sum_{i=0}^{l_1} E[T^i | S^0] \tag{7.2}$$

All the customers that will enter service before our customer are type 1 customers. The number of those customers is known. Therefore, the only difference between the accurate average of waiting time and the approximated waiting time is caused by errors in computing average iteration times (that is affected both by service configuration probabilities and by the exponential approximation). If accurate calculations for average iteration times were used, then the approximated mean waiting time is accurate.

We will now analyze the impact of calculating the average iteration time using exponential approximation, on the estimated waiting time in the above case.

In general, when $i \to \infty$ both the accurately calculated average of iteration time $E[T^i|S^0]$ and the approximated value $\tilde{E}[T^i|S^0]$ approach $\frac{1}{2\mu_1}$, and are independent of the initial state S^0.
Therefore, when $l_1 \to \infty$ we get:

$$\lim_{l_1 \to \infty} \frac{E[W_1(S^0; l_1, \cdot)]}{\tilde{W_1}(S^0; l_1, \cdot)} = 1.$$

We will now discuss each of three possible cases of S^0 in more details.

i. $\mathbf{S^0} = (\mathbf{2}, \mathbf{0})$; two type 1 customers are being served upon arrival

In this case the approximated iteration time equals the real value of the average iteration time. Therefore, the approximation is accurate.

$$\begin{aligned} E[W_1(2,0; l_1, \cdot)] = \sum_{i=0}^{l_1} E(\frac{1}{2\mu_1}) &= \frac{l_1 + 1}{2\mu_1} \\ \tilde{E}[W_1(2,0; l_1, \cdot)] = \sum_{i=0}^{l_1} (\frac{1}{2\mu_1}) &= \frac{l_1 + 1}{2\mu_1} \end{aligned}$$

For any value of l_1 we, therefore, get:

$$\begin{aligned} E[W_1(2,0; l_1, \cdot)] - \tilde{E}[W_1(2,0; l_1, \cdot)] &= 0 \\ \frac{E[W_1(2,0; l_1, \cdot)]}{\tilde{E}[W_1(2,0; l_1, \cdot)]} &= 1 \end{aligned}$$

ii. $\mathbf{S_0} = (\mathbf{1}, \mathbf{1})$; one customer of each type is being served upon arrival

The service configuration in iteration i S^i is $(1,1)$ if the type 2 customer that was in service upon arrival, is still being served. The probability for that is $(\frac{\mu_1}{\mu_1 + \mu_2})^i$. We have already defined in (4.1) $\frac{\mu_1}{\mu_1 + \mu_2} = q$. The average time of iteration i is:

$$E[T^i|S^0 = (1,1)] = \frac{1 - q^i}{2\mu_1} + \frac{q^i}{\mu_1 + \mu_2}.$$

Assume, that server 2 was the one that served the type 2 customer. Then, by the approximation server 1 service rate is still μ_1. Server's 2 service rate is computed by $\frac{1}{\substack{average\ service\ time \\ of\ server\ 2}}$. The overall service rate in the system is the sum of service rates of the two servers, and the average iteration time is the inverse of the service rate. Therefore, approximated iteration time is given by:

$$\tilde{E}[T^i|S^0 = (1,1)] = \frac{1}{\frac{\mu_1 \cdot \mu_2}{\mu_2 - \mu_2 q^i + \mu_1 q^i} + \mu_1}$$

Summing iteration times over all $l_1 + 1$ iterations, we obtain the accurate and the approximated values of the average waiting time:

$$\begin{aligned}
E[W_1(1,1;l_1,\cdot)] &= \sum_{i=0}^{l_1} \frac{1-q^i}{2\mu_1} + \frac{q^i}{\mu_1+\mu_2} \\
\tilde{E}[W_1(1,1;l_1,\cdot)] &= \sum_{i=0}^{l_1} \frac{1}{\frac{\mu_1\mu_2}{\mu_2-\mu_2 q^i+\mu_1 q^i} \ + \ \mu_1}
\end{aligned}$$

As i increases, the approximated iteration time approaches the accurate average of iteration time:

$$\begin{aligned}
\lim_{i\to\infty} E[T^i] &= \lim_{i\to\infty} \frac{1-q^i}{2\mu_1} + \frac{q^i}{\mu_1+\mu_2} = \frac{1}{2\mu_1} \\
\lim_{i\to\infty} \tilde{E}[T^i] &= \lim_{i\to\infty} \frac{1}{\frac{\mu_1\mu_2}{\mu_2-\mu_2 q^i+\mu_1 q^i} \ + \ \mu_1} = \frac{1}{2\mu_1}.
\end{aligned} \tag{7.3}$$

As l_1 increases, the total estimated waiting time becomes more accurate:

$$\begin{aligned}
&\lim_{l_1\to\infty} \frac{E[W_1(1,1;l_1,\cdot)] - \tilde{E}[W_1(1,1;l_1,\cdot)]}{l_1} \\
&= \lim_{l_1\to\infty} \frac{1}{l_1} \cdot \sum_{i=0}^{l_1} \left(\frac{1-q^i}{2\mu_1} + \frac{q^i}{\mu_1+\mu_2} - \frac{1}{\frac{\mu_1\mu_2}{\mu_2-\mu_2 q^i+\mu_1 q^i} \ + \ \mu_1} \right) \\
&= 0, \\
&\lim_{l_1\to\infty} \frac{E[W_1(1,1;l_1,\cdot)]}{\tilde{E}[W_1(1,1;l_1,\cdot)]} = \lim_{l_1\to\infty} \frac{\frac{E[W_1(1,1;l_1,\cdot)]}{l_1}}{\frac{\tilde{E}[W_1(1,1;l_1,\cdot)]}{l_1}} = \lim_{i\to\infty} \frac{E[T^i]}{\tilde{E}[T^i]} \\
&= 1.
\end{aligned}$$

iii. $\mathbf{S_0 = (0,2)}$; two type 2 customers are being served upon arrival

Similar to the previous case, the average time of iteration i, $(i > 1)$ is:

$$E[T^i | S^0 = (0,2)] = \frac{1-q^{i-1}}{2\cdot\mu_1} + \frac{q^{i-1}}{\mu_1+\mu_2},$$

while the approximated time of the same iteration is:

$$\tilde{E}[T^i | S^0 = (0,2)] = \frac{1}{\frac{\mu_1\cdot\mu_2}{\mu_2-\mu_2\cdot q^{i-1}+\mu_1\cdot q^{i-1}} \ + \ \mu_1}. \tag{7.4}$$

The average time of first iteration obtained by the approximation is accurate and equals $\frac{1}{2\mu_2}$.

The accurate value of the average waiting time is:

$$E[W_1(0,2;l_1,\cdot)] \;=\; \frac{1}{2\cdot\mu_2}+\sum_{i=1}^{l_1}(\frac{1-q^{i-1}}{2\cdot\mu_1}+\frac{q^{i-1}}{\mu_1+\mu_2})$$

The approximated value of average waiting time is:

$$\tilde{E}[W_1(0,2;l_1,\cdot)] \;=\; \frac{1}{2\cdot\mu_2}+\sum_{i=1}^{l_1}(\frac{1}{\frac{\mu_1\cdot\mu_2}{\mu_2-\mu_2\cdot q^{i-1}+\mu_1\cdot q^{i-1}}\;+\;\mu_1})$$

Again, when l_1 is big enough, the approximation yields good results:

$$\begin{aligned}\lim_{l_1\to\infty}\frac{E[W_1(0,2;l_1,\cdot)]}{\tilde{E}[W_1(0,2;l_1,\cdot)]} &= \lim_{l_1\to\infty}\frac{\frac{E[W_1(0,2;l_1,\cdot)]}{l_1}}{\frac{\tilde{E}[W_1(0,2;l_1,\cdot)]}{l_1}}\\ &= 1.\end{aligned}$$

As shown, inaccuracies resulting from approximating the hyper-exponential service times by exponential service times decrease as the queue size (L_1^0) increases.

7.2 Estimating the waiting times of low priority customers - number of iterations

Waiting time approximation for low priority customers involves several sources of inaccuracies including:

- Approximating hyper-exponential service times by exponential service times. It is possible, however, to use exact calculation by a recursive procedure, see Appendix A for documentation.
- Approximating number of type 1 arrivals by its average.

All calculations in this section are done under the assumption that iteration times and service configuration probabilities are accurately calculated and not being approximated by exponential variables. Assuming that, we isolate the second source of inaccuracy and we only evaluate the affect of replacing the actual number of type 1 arrivals with the average number of arrivals.

The waiting time of a type 2 customer who finds l_1 type 1 and l_2 type 2 customers in queue, and a service configuration of S^0 consists of l_2+1 periods:

i. U^0; The first period is the time since arrival and until the first type 2 customer enters service.

ii. $U^i, i = 1, .., l_2$; Additional l_2 periods, each of them includes the time between successive entries of type 2 customers to service.

$U^i; i \geq 1$ is the time between service entries of the $i^{'th}$ type 2 customer and the $(i+1)^{'th}$ type 2 customer. Actually U^i is the equivalent to B^i in Chapter 4. We will estimate the approximation accuracy for each of the two types of periods, and then combine the results to conclude the accuracy of the total waiting time estimation.

7.2.1 $\mathbf{U^0}$; Time until the first type 2 customer enters service

The time until the first type 2 customer enters service is equal in distribution to the waiting time of a type 1 customer who finds $l_1 + A_1^0$ type 1 customers in line, where A_1^0 is the number of type 1 customers that arrived during the waiting time and passed our customer in line. Formally:

$$U^0 \overset{d}{=} W_1(s_1, s_2; l_1 + A_1^0, \cdot),$$

when

$$A_1^0 \overset{d}{=} Poisson\left(\lambda_1 \cdot U^0\right). \tag{7.5}$$

We now analyze the effect of replacing A_1^0 with its average for each of the 3 different initial service configurations. We sometimes use $A_1(s_1, s_2)$ as a shorter form of $A_1^0|S^0 = (s_1, s_2)$

i. $\mathbf{S^0 = (2,0)}$; Two type 1 customers are being served

$U^0|S^0 = (2,0)$ is a busy period, in which the service time of the first customer is $Erlang(l_1 + 1,\ 2\mu_1)$, and all other (arriving) customers have exponential $(2\mu_1)$ service times (or to $l_1 + 1$ standard busy periods).
The average time $E[U^0]$ is:

$$E[U^0|S^0 = (2,0)] = \frac{l_1 + 1}{2\mu_1 - \lambda_1}.$$

$\tilde{E}[U^0|S^0 = (2,0)]$ is the approximated mean of of the period, obtained by replacing the number of arrivals with its average $\tilde{E}[A_1^0|S^0 = (2,0)]$:

$$\tilde{E}[U^0|S^0 = (2,0)] = \sum_{i=0}^{l_1 + \tilde{E}[A_1(2,0)]} \frac{1}{2\mu_1}$$

$$
\begin{aligned}
&= \frac{l_1 + 1 + \tilde{E}[A_1(2,0)]}{2\mu_1} \\
&= \frac{l_1 + 1 + \lambda_1 \cdot \tilde{E}[U^0|S^0 = (2,0)]}{2\mu_1}.
\end{aligned}
$$

Therefore,

$$
\tilde{E}[U^0|S^0 = (2,0)] \cdot (1 - \frac{\lambda_1}{2\mu_1}) = \frac{l_1 + 1}{2\mu_1},
$$

and

$$
\tilde{E}[U^0|S^0 = (2,0)] = \frac{l_1 + 1}{2\mu_1 - \lambda_1}.
$$

The approximated value is accurate in this case.

ii. $\mathbf{S^0 = (1,1)}$; One customer of each type is being served

The accurate value for the mean time of the i'th iteration is:

$$
\begin{aligned}
E[T^i|S^0 = (1,1)] &= \tilde{E}[T^i|S^0 = (1,1)] \\
&= \left(\frac{q^{i-1}}{\mu_1 + \mu_2} + \frac{1 - q^{i-1}}{2\mu_1}. \right),
\end{aligned}
$$

The average duration of period $U^0|S^0 = (1,1)$ is

$$
E[U^0|S^0 = (1,1)] = E\left[\sum_{i=0}^{l_1 + A_1(1,1)} [\frac{q^i}{\mu_1 + \mu_2} + \frac{1 - q^i}{2\mu_1}] \right].
$$

Assuming that the exact means of iteration times are used, the approximated value $\tilde{E}[U^0|S^0 = (1,1)]$ is:

$$
\tilde{E}[U^0|S^0 = (1,1)] = \sum_{i=0}^{l_1 + \tilde{E}[A_1(1,1)]} [\frac{q^i}{\mu_1 + \mu_2} + \frac{1 - q^i}{2\mu_1}].
$$

The above expressions are easily developed into:

$$
\begin{aligned}
&E[U^0|S^0 = (1,1)] \cdot (1 - \frac{\lambda_1}{2\mu_1}) \\
&= \frac{1 - q^{l_1+1} \cdot E[q^{A_1^0}|S^0 = (1,1)]}{\mu_2} + \frac{l_1 + 1}{2\mu_1} \\
&\quad - \frac{1 - q^{l_1+1} \cdot E[q^{A_1^0}|S^0 = (1,1)]}{2\mu_1 \cdot \mu_2} \cdot (\mu_1 + \mu_2) \qquad (7.6)
\end{aligned}
$$

and

$$\begin{aligned}
&\tilde{E}[U^0|S^0 = (1,1)] \cdot (1 - \frac{\lambda_1}{2\mu_1}) \\
&= \quad \frac{1 - q^{l_1+1} \cdot q^{\tilde{E}[A_1(1,1)]}}{\mu_2} + \frac{l_1 + 1}{2\mu_1} - \frac{1 - q^{l_1+1} \cdot q^{\tilde{E}[A_1(1,1)]}}{2\mu_1 \cdot \mu_2}(\mu_1 + \mu_2). \qquad (7.7)
\end{aligned}$$

We now let l_1 approach ∞. Since $q \in (0,1)$, both $E[q^{A_1^0}|S^0 = (1,1)]$ and $q^{\tilde{E}[A_1^0|S^0=(1,1)]}$ are always finite. As in the previous case, when l_1 is large enough the approximated value approaches the accurate value:

$$\begin{aligned}
\lim_{l_1 \to \infty} \frac{E\left[U^0|S^0 = (1,1)\right]}{\tilde{E}\left[U^0|S^0 = (1,1)\right]} &= \lim_{l_1 \to \infty} \frac{E\left[U^0|S^0 = (1,1)\right] \cdot (\frac{2\mu_1 - \lambda_1}{2\mu_1})}{\tilde{E}\left[U^0|S^0 = (1,1)\right] \cdot (\frac{2\mu_1 - \lambda_1}{2\mu_1})} \\
&= 1.
\end{aligned}$$

Combining (7.6) and (7.7), we can see that the difference between the accurate and the approximated means is a function of $q^{\tilde{E}[A_1(1,1)]} - E[q^{A_1(1,1)}]$:

$$\begin{aligned}
&\Big[E[U^0|S^0 = (1,1)] - \tilde{E}[U^0|S^0 = (1,1)]\Big] \cdot (\frac{2\mu_1 - \lambda_1}{2\mu_1}) \\
&= \quad \frac{q^{l_1+1}(q^{\tilde{E}[A_1(1,1)]} - E[q^{A_1(1,1)}])}{\mu_2} - \frac{q^{l_1+1}(q^{\tilde{E}[A_1(1,1)]} - E[q^{A_1(1,1)}])}{2\mu_1 \cdot \mu_2}(\mu_1 + \mu_2) \\
&= \quad (q^{\tilde{E}[A_1^0(1,1)]} - E[q^{A_1(1,1)}]) \cdot q^{l_1+1} \cdot \left[\frac{1}{\mu_2} - \frac{\mu_1 + \mu_2}{2\mu_1 \cdot \mu_2}\right] \\
&= \quad (q^{\tilde{E}[A_1(1,1)]} - E[q^{A_1(1,1)}]) \cdot q^{l_1} \cdot \frac{1}{\mu_2} \cdot \left[q - \frac{1}{2}\right].
\end{aligned}$$

Based on the above results we would like to note the following points:

- When $A_1^0|S^0 = (1,1)$ and $\tilde{E}[A_1^0|S^0 = (1,1)]$ are large, both $E[q^{A_1^0}|S^0 = (1,1)]$, and $q^{\tilde{E}[A_1^0|S^0=(1,1)]}$ approach 0, and so does the difference between the approximated and the average periods.
- The expression $(q^{\tilde{E}[A_1^0|S^0=(1,1)]} - E[q^{A_1}|S^0 = (1,1)])$ is bounded in the range $(-1,1)$, and therefore an upper-bound for the difference between the accurate average waiting time and the approximated waiting time is:

  $$\left|\Big[E[U^0|S^0 = (1,1)] - \tilde{E}[U^0|S^0 = (1,1)]\Big] \cdot (\frac{2\mu_1 - \lambda_1}{2\mu_1})\right| \quad \leq \quad \left|\frac{q^{l_1}}{\mu_2}(q - \frac{1}{2})\right|$$

  From the above result we can see that if $\mu_1 = \mu_2$, which implies $q = \frac{1}{2}$, the approximated mean of the waiting time is accurate. We can also see that as the number of type 1 customers waiting upon arrival increases, the difference between the accurate mean of the waiting time and the approximated one is decreases.

iii. $\mathbf{S_0 = (0, 2)}$; two type 2 customers are being served

Assuming $l_1 > 0$ the average time until the first type 2 customer enters service is:

$$E[U^0|S_0 = (0,2)] \;=\; \frac{1}{2\mu_2} + E\left[\sum_{i=0}^{l_1+A_1^0-1}(\frac{q^i}{\mu_1+\mu_2} + \frac{1-q^i}{2\mu_1})|S_0 = (0,2)\right],$$

and

$$\begin{aligned}&E[U^0|S_0 = (0,2)] \cdot (1 - \frac{\lambda_1}{2\mu_1})\\ =\;& \frac{1}{2\mu_2} + \frac{l_1}{2\mu_1} + \frac{\mu_1-\mu_2}{2\mu_1\cdot(\mu_1+\mu_2)} \cdot \left(\frac{1-q^{l_1}E[q^{A_1(0,2)}]}{1-q}\right).\end{aligned} \tag{7.8}$$

When replacing the number of type 1 arrivals with its average, we obtain:

$$\tilde{E}\left[U^0|S_0 = (0,2)\right] \;=\; \frac{1}{2\mu_2} + \sum_{i=0}^{l_1+\tilde{E}[A_1(0,2)]-1}[\frac{q^i}{\mu_1+\mu_2} + \frac{1-q^i}{2\mu_1}]$$

and

$$\begin{aligned}&\tilde{E}\left[U^0|S_0 = (0,2)\right] \cdot (1 - \frac{\lambda_1}{2\mu_1})\\ =\;& \frac{1}{2\mu_2} + \frac{l_1}{2\mu_1} + \frac{\mu_1-\mu_2}{2\mu_1\cdot(\mu_1+\mu_2)} \left(\frac{1-q^{l_1}\cdot q^{\tilde{E}[A_1(0,2)]}}{1-q}\right).\end{aligned} \tag{7.9}$$

The difference between the accurate value in (7.8) and the approximated one in (7.9) is:

$$\begin{aligned}&\Big[E[U^0|S_0 = (0,2)] - \tilde{E}[U^0|S_0 = (0,2)]\Big] \cdot \left(\frac{2\mu_1-\lambda_1}{2\mu_1}\right)\\ =\;& q^{l_1}\left(q^{\tilde{E}[A_1^0(0,2)]} - E[q^{A_1(0,2)}]\right) \cdot \frac{\mu_1-\mu_2}{2\cdot\mu_1\cdot\mu_2}.\end{aligned}$$

Since $\left[q^{\tilde{E}[A_1^0|S^0=(0,2)]} - E[q^{A_1}|S^0 = (0,2)]\right] \leq 1$, an upper bound for the above expression is denoted by:

$$\left|\Big[E[U^0|S_0 = (0,2)] - \tilde{E}[U^0|S_0 = (0,2)]\Big] \cdot (\frac{2\mu_1-\lambda_1}{2\mu_1})\right| \;\leq\; \left|q^{l_1}\frac{\mu_1-\mu_2}{2\cdot\mu_1\cdot\mu_2}\right|.$$

In this case as well, when the number of type 1 waiting customers, l_1, is big, the approximated value of U^1 approaches the accurate value:

$$\lim_{l_1\to\infty}[E[U^0|S_0 = (0,2)] - \tilde{E}[U^0|S_0 = (0,2)]\left(\frac{2\mu_1-\lambda_1}{\mu_2}\right) \;=\; 0$$

and

$$\lim_{l_1 \to \infty} \frac{E[U^0|S_0 = (0,2)]}{\tilde{E}[U^0|S_0 = (0,2)]} \quad = \quad 1.$$

In addition, here again, when $\mu_1 = \mu_2$ the approximated value is accurate.

iv. $\mathbf{U^0}$: The time until the first type 2 customer enters service - Summary

In all of the above 3 cases, the average time until the first type 2 customer enters service is of the form:

$$E[U^0|S^0; L^0 = (l_1, \cdot)] \cdot (1 - \frac{\lambda_1}{2\mu_1}) \quad = \quad C_1 + C_2 \cdot l_1 + C_3 \cdot q^{l_1} E[q^{A_1}|S^0],$$

when C_1, C_2, C_3 are known. The value obtained by approximating number of arrivals is of the form:

$$\tilde{E}[U^0|S^0; L^0 = (l_1, \cdot)] \cdot (1 - \frac{\lambda_1}{2\mu_1}) \quad = \quad C_1 + C_2 \cdot l_1 + C_3 \cdot q^{l_1} q^{\tilde{E}[A_1|S^0]}.$$

If there are no type 2 customers waiting in queue upon arrival ($L_2^0 = 0$), then the total waiting time is denoted by the above results. A few conclusions can be made for this case:

- We have found an upper bound for the difference between the accurate and the approximated values of the average waiting time.
- When the number of type 1 customers waiting upon arrival is very large, the approximated waiting times, obtained by replacing the number of arrivals with its average, are of high accuracy.
- When service rates of the two service types are the same, the approximated waiting times, obtained by replacing the number of arrivals with its average, are accurate.

We now evaluate the accurate and the approximated values of the mean time from the moment when the first type 2 customer enters service until our customer enters service. This is relevant for cases when upon arrival, there are $l_2 > 0$ type 2 customers in queue.

7.2.2 $\mathbf{W_2(S^0; L^0)}$ - Time between successive entries of type 2 customers

Recall that U^i is the time between service entries of the i'th and the $(i+1)$'th customers. S^{U^i} stands for the service configuration at the beginning of period U^i. The mean of the total waiting time of a type 2 customer is:

$$\begin{aligned}
&E[W_2(s_1, s_2; l_1, l_2)] \\
&= \sum_{i=0}^{l_2} E[U^i|S^0] \\
&= E[U^0|S^0] + \sum_{i=1}^{l_2} E[U^i|S^{U^i} = (1,1)] \cdot P[S^{U^i} = (1,1)|S^0] \\
&\quad + \sum_{i=1}^{l_2} E[U^i|S^{U^i} = (0,2)] \cdot P[S^{U^i} = (0,2)|S^0].
\end{aligned}$$

Similarly, the approximated value is:

$$\begin{aligned}
&\tilde{E}[W_2(s_1, s_2; l_1, s_2)] \\
&= \sum_{i=0}^{l_2} \tilde{E}[U^i|S^0] \\
&= \tilde{E}[U^0|S^0] + \sum_{i=1}^{l_2} \tilde{E}[U^i|S^{U^i} = (1,1)] \cdot \tilde{P}[S^{U^i} = (1,1)|S^0] \\
&\quad + \sum_{i=1}^{l_2} \tilde{E}[U^i|S^{U^i} = (0,2)] \cdot \tilde{P}[S^{U^i} = (0,2)|S^0].
\end{aligned}$$

We now look into the accurate and approximated values of $E[U^i]$, $i > 0$, conditioning on the service configuration at the beginning of the period, S^{U^i}.

i. Approximated and accurate mean of U^i where $\mathbf{S^{U^i}} = (\mathbf{1}, \mathbf{1})$

$A_1^{U^i}$ is the number of type 1 customers that arrive during period U^i. $A_1^{U^i}(s_1, s_2)$ is the number of type 1 customers that arrive during a period, given that the service configuration at the beginning of the period is (s_1, s_2). For convenience, we sometimes use $A_1(s_1, s_2)$ (without period indication) to denote the number of arrivals (type 1) during some busy period U^i.

The mean time of period U^i provided that service configuration at the beginning of the period is $S^{U^i} = (1,1)$ is:

$$E[U^i \mid (S^{U^i} = (1,1))] \;=\; E\left[\sum_{i=0}^{A_1^{U^i}|S^{U^i}=(1,1)} [\frac{q^i}{\mu_1 + \mu_2} + \frac{1-q^i}{2\mu_1}]\right], \qquad (7.10)$$

and

$$\begin{aligned} & E[U^i|S^{U^i}=(1,1)] \cdot \left(1-\frac{\lambda_1}{2\mu_1}\right) \\ = \;& \frac{1}{2\mu_1}+\frac{\mu_1-\mu_2}{2\mu_1\mu_2}\left(1-Eq^{A^{U^i|S^{U^i}=(1,1)}+1}\right) \\ = \;& \frac{1}{2\mu_2}-\frac{\mu_1-\mu_2}{2\mu_1\cdot\mu_2}E[q^{A_1^{U^i}+1}|S^{U^i}=(1,1)]. \end{aligned} \tag{7.11}$$

When replacing the number of arrivals in (7.10) and in (7.11) with its average we obtain the following equation:

$$\begin{aligned} & \tilde{E}[U^i|S^{U^i}=(1,1)] \cdot \left(1-\frac{\lambda_1}{2\mu_1}\right) \\ = \;& \frac{1}{2\mu_1}+\left(1-q^{\tilde{E}(A_1^{U^i}|S^{U^i}=(1,1)+1)}\right)\cdot\left(\frac{\mu_1-\mu_2}{2\mu_1\mu_2}\right). \end{aligned}$$

ii. Approximated and accurate mean of U^i where $\mathbf{S^{U^i}=(0,2)}$

The expected time between successive entries of type 2 customers to service is:

$$\begin{aligned} & E\left[U^i|S^{U^i}=(0,2)\right] \\ = \;& \frac{1}{2\mu_2}+P\left(A_1^{U^i}>0|S^{U^i}=(0,2)\right)\cdot\sum_{i=0}^{A_1^{U^i}(0,2)-1|A_1^{U^i}(0,2)>0}[\frac{q^i}{\mu_1+\mu_2}+\frac{1-q^i}{2\mu_1}] \\ = \;& \frac{1}{2\mu_2}+\frac{\lambda_1}{\lambda_1+2\mu_2}\sum_{i=0}^{A_1^{U^i}(0,2)-1|A_1^{U^i}>0}\left[\frac{q^i}{\mu_1+\mu_2}+\frac{1-q^i}{2\mu_1}\right] \\ = \;& \frac{1}{2\mu_2}+\frac{\lambda_1}{\lambda_1+2\mu_2}\cdot\frac{\mu_1-\mu_2}{2\mu_1(\mu_1+\mu_2)}\cdot E\left[q^{A_1|A_1>0,S^{U^i}=(0,2)}\right] \\ & +\frac{\lambda_1}{\lambda_1+2\mu_2}\cdot E\left[A_1^{U^i}-1|A_1^{U^i}>0,S^{U^i}=(0,2)\right]\frac{1}{2\mu_1} \end{aligned} \tag{7.12}$$

Since

$$\begin{aligned} E\left[q^{A_1^{U^i}}|A_1^{U^i}>0,S^{U^i}=(0,2)\right] &= \frac{E[q^{A_1(0,2)}]-P(A_1(0,2)=0)}{\frac{\lambda_1}{\lambda_1+2\mu_2}} = \\ &= \frac{E[q^{A_1(0,2)}]-\frac{2\mu_2}{\lambda_1+2\mu_2}}{\frac{\lambda_1}{\lambda_1+2\mu_2}} \end{aligned}$$

and

$$E[A_1^{U^i}|A_1^{U^i}>0,S^{U^i}=(0,2)] = \frac{E[A_1(0,2)]}{\frac{\lambda_1}{\lambda_1+2\mu_2}},$$

the average time between successive service entries of type 2 customers, denoted in (7.12), can also be expressed by:

$$\begin{aligned} E\left[U^i|S^{U^i}=(0,2)\right] &= \frac{1}{2\mu_2}+\frac{\lambda_1}{\lambda_1+2\mu_2}\cdot\frac{\mu_1-\mu_2}{2\mu_1\mu_2} \\ &\quad -\frac{\mu_1-\mu_2}{2\mu_1(\mu_1+\mu_2)}\cdot\left[E\left[q^{A_1}|S^{U^i}=(0,2)\right]-\frac{2\mu_2}{\lambda_1+2\mu_2}\right] \\ &\quad +E\left[A_1|S^{U^i}=(0,2)\right]\cdot\frac{1}{2\mu_1}. \end{aligned}$$

Replacing the number of arrivals by its average, we get:

$$\begin{aligned} \tilde{E}\left[U^i|S^{U^i}=(0,2)\right] &= \frac{1}{2\mu_2}+\frac{\lambda_1}{\lambda_1+2\mu_2}\cdot\frac{\mu_1-\mu_2}{2\mu_1\mu_2} \\ &\quad -\frac{\mu_1-\mu_2}{2\mu_1(\mu_1+\mu_2)}\cdot\left[q^{\tilde{E}[A_1^{U^i}]}-\frac{2\mu_2}{\lambda_1+2\mu_2}|S^{U^i}=(0,2)\right] \\ &\quad +\left[\tilde{E}[A_1^{U^i}-1|S^{U^i}=(0,2)]\right]\cdot\frac{1}{2\mu_1}. \end{aligned}$$

We can use the following connections

$$\begin{aligned} E[A_1^{U^i}|S^{U^i}=(0,2)] &= \lambda_1 E[U^i|S^{U^i}=(0,2)] \\ \tilde{E}[A_1^{U^i}|S^{U^i}=(0,2)] &= \lambda_1 \tilde{E}[U^i|S^{U^i}=(0,2)], \end{aligned}$$

to conclude that

$$\begin{aligned} & E[U^i|S^{U^i}=(0,2)]\cdot(1-\frac{\lambda_1}{2\mu_1}) \\ = \;& \frac{1}{2\mu_2}+\frac{\lambda_1}{\lambda_1+2\mu_2}\cdot\frac{\mu_1-\mu_2}{2\mu_1\mu_2} \\ & -\frac{\mu_1-\mu_2}{2\mu_1(\mu_1+\mu_2)}\cdot E\left[q^{A_1^{U^i}}-\frac{2\mu_2}{\lambda_1+2\mu_2}|S^{U^i}=(0,2)\right] \qquad (7.13) \\ & \tilde{E}[U^i|S^{U^i}=(0,2)]\cdot(1-\frac{\lambda_1}{2\mu_1}) \\ = \;& \frac{1}{2\mu_2}+\frac{\lambda_1}{\lambda_1+2\mu_2}\cdot\frac{\mu_1-\mu_2}{2\mu_1\mu_2} \\ & -\frac{\mu_1-\mu_2}{2\mu_1(\mu_1+\mu_2)}\cdot\left[q^{\tilde{E}A_1^{U^i}}-\frac{2\mu_2}{\lambda_1+2\mu_2}|S^{U^i}=(0,2)\right]. \qquad (7.14) \end{aligned}$$

7.2.3 $\sum_{i=0}^{l_2} U^i, i \geq 1$ - Total waiting time of a type 2 customer

Summarizing all the above and simplifying expressions we obtain a general form for the waiting time of a type 2 customer:

$$\begin{aligned}
&E[W_2(s_1, s_2; l_1, s_2)] \cdot (1 - \frac{\lambda_1}{2\mu_1}) \\
&= \quad C_1 + C_2 \cdot l_1 + C_3 q^{l_1} E[q^{A_1^0|S^0}] \\
&\quad + \frac{1}{2\mu_2} l_2 + \frac{2\lambda_1 + 2\mu_2}{\lambda_1 + 2\mu_2} \cdot \frac{\mu_1 - \mu_2}{2\mu_1\mu_2} \sum_{i=1}^{l_2} P[S^{U^i} = (0,2)] \\
&\quad - \frac{\mu_1 - \mu_2}{2\mu_1\mu_2} \sum_{i=1}^{l_2} E[q^{A_1^{U^i}} | S^{U^i} = (1,1)] \cdot P(S^{U^i} = (1,1)|S^0) \\
&\quad - \frac{\mu_1 - \mu_2}{2\mu_1\mu_2} \sum_{i=1}^{l_2} E[q^{A_1^{U^i}} | S^{U^i} = (0,2)] \cdot P(S^{U^i} = (0,2)|S^0) \\
&= \quad C_1 + C_2 \cdot l_1 + C_3 \cdot q^{l_1} \cdot E[q^{A_1^0|S^0}] \\
&\quad + \frac{1}{2\mu_2} l_2 + \frac{2\lambda_1 + 2\mu_2}{\lambda_1 + 2\mu_2} \cdot \frac{\mu_1 - \mu_2}{2\mu_1\mu_2} \sum_{i=1}^{l_2} P[S^{U^i} = (0,2)] \\
&\quad - \frac{\mu_1 - \mu_2}{2\mu_1\mu_2} \sum_{i=1}^{l_2} E[q^{A_1^{U^i}}]. \qquad (7.15)
\end{aligned}$$

By replacing $E[q^{A_1}]$ with $q^{\tilde{E}[A_1]}$ we get:

$$\begin{aligned}
&\tilde{E}[W_2(s_1, s_2; l_1, s_2)] \cdot (1 - \frac{\lambda_1}{2\mu_1}) \\
&= \quad C_1 + C_2 \cdot l_1 + C_3 \cdot q^{l_1} \cdot q^{\tilde{E}[A_1^0|S^0]} \\
&\quad + \frac{1}{2\mu_2} l_2 + \frac{2\lambda_1 + 2\mu_2}{\lambda_1 + 2\mu_2} \cdot \frac{\mu_1 - \mu_2}{2\mu_1\mu_2} \sum_{i=1}^{l_2} \tilde{P}[S^{U^i} = (0,2)] \\
&\quad - \frac{\mu_1 - \mu_2}{2\mu_1\mu_2} \sum_{i=1}^{l_2} q^{\tilde{E}A_1^{U^i}} \qquad (7.16)
\end{aligned}$$

The difference between the expressions in (7.15) and in (7.16) and the ratio between them depend on the differences in the following values (and their limits when $i \to \infty$):

- $E[q^{A_1^{U^i}} | S^{U^i} = (1,1)]$ vs. $q^{\tilde{E}A_1^{U^i} | S^{U^i} = (1,1)}$
- $E[q^{A_1^{U^i}} | S^{U^i} = (0,2)]$ vs. $q^{\tilde{E}A_1^{U^i} | S^{U^i} = (0,2)}$
- $P[S^{U^i} = (1,1)|S^0]$ vs. $\tilde{P}[S^{U^i} = (1,1)|S^0]$

All the above expressions are determined by μ_1, μ_2 and λ_1. These parameters, as well as the initial state of the system (S^0, L^0), determine the accuracy of the approximation.

Following we analyze the value of the above ratio and of the difference between (7.15) and (7.16) under several circumstances.

i. When $\mu_1 = \mu_2 = \mu$, the expressions in (7.15) and in (7.16) simplify to:

$$\begin{aligned} E[W_2(s_1,s_2;l_1,s_2)]\cdot(1-\frac{\lambda_1}{2\mu_1}) &= C_1 + C_2\cdot l_1 + C_3\cdot q^{l_1}\cdot E[q^{A_1^0|S^0}] + \frac{1}{2\mu_2}l_2 \\ \tilde{E}[W_2(s_1,s_2;l_1,s_2)]\cdot(1-\frac{\lambda_1}{2\mu_1}) &= C_1 + C_2\cdot l_1 + C_3\cdot q^{l_1}\cdot q^{\tilde{E}[A_1^0|S^0]} + \frac{1}{2\mu_2}l_2 \end{aligned}$$

The difference between the above two expression is given by:

$$\begin{aligned} &\Big(E[W_2(s_1,s_2;l_1,s_2)] - \tilde{E}[W_2(s_1,s_2;l_1,s_2)]\Big)\cdot(1-\frac{\lambda_1}{2\mu}) \\ &= \quad C_3\cdot q^{l_1}\cdot(E[q^{A_1^0}|S^0] - q^{\tilde{E}[A_1^0]|S^0}). \end{aligned}$$

The above difference is bounded by:

$$\begin{aligned} \Big|E[W_2(s_1,s_2;l_1,s_2)] - \tilde{E}[W_2(s_1,s_2;l_1,s_2)]\Big|\cdot(1-\frac{\lambda_1}{2\mu}) &\leq \Big|C_3\cdot q^{l_1}\Big| \\ &= \Big|C_3\cdot(\frac{1}{2})^{l_1}\Big|. \end{aligned}$$

ii. When l_1 is large enough we obtain the following limits:

$$\begin{aligned} \lim_{l_1\to\infty}\frac{E[W_2(s_1,s_2;l_1,s_2)]\cdot(1-\frac{\lambda_1}{2\mu_1})}{l_1} &= C_2 \\ \lim_{l_1\to\infty}\frac{\tilde{E}[W_2(s_1,s_2;l_1,s_2)]\cdot(1-\frac{\lambda_1}{2\mu_1})}{l_1} &= C_2 \end{aligned}$$

The ratio between the accurate mean of waiting time and the approximated one approaches 1:

$$\lim_{l_1\to\infty}\frac{E[W_2(s_1,s_2;l_1,s_2)]}{\tilde{E}[W_2(s_1,s_2;l_1,s_2)]} = 1. \tag{7.17}$$

iii. When l_2 increases, the following limit is obtained for the the accurate waiting time:

$$\lim_{l_2\to\infty}\frac{E[W_2(s_1,s_2;l_1,s_2)](1-\frac{\lambda_1}{2\mu_1})}{l_1\cdot l_2}$$

$$
\begin{aligned}
&= \frac{1}{2\mu_2 \cdot l_1} + \frac{2\lambda_1 + 2\mu_2}{\lambda_1 + 2\mu_2} \cdot \frac{\mu_1 - \mu_2}{2\mu_1 \cdot \mu_2} \cdot \frac{\lim_{i\to\infty} P(S^{U^i} = (0,2))}{l_1} \\
&\quad - \frac{\mu_1 - \mu_2}{2\mu_1 \cdot \mu_2} \cdot \frac{\lim_{i\to\infty} Eq^{A_1^{U^i}}}{l_1}. \qquad (7.18)
\end{aligned}
$$

The limit of the approximated waiting time is:

$$
\begin{aligned}
&\lim_{l_2\to\infty} \frac{\tilde{E}[W_2(s_1, s_2; l_1, s_2)](1 - \frac{\lambda_1}{2\mu_1})}{l_1 \cdot l_2} \\
&= \frac{1}{2\mu_2 \cdot l_1} + \frac{2\lambda_1 + 2\mu_2}{\lambda_1 + 2\mu_2} \cdot \frac{\mu_1 - \mu_2}{2\mu_1 \cdot \mu_2} \cdot \frac{\lim_{i\to\infty} \tilde{P}(S^{U^i} = (0,2))}{l_1} \\
&\quad - \frac{\mu_1 - \mu_2}{2\mu_1 \cdot \mu_2} \cdot \frac{\lim_{i\to\infty} q^{\tilde{E}A_1^{U^i}}}{l_1}. \qquad (7.19)
\end{aligned}
$$

Since $P(S^{U^i} = (0,2))$, $Eq^{A_1^{U^i}}$, $\tilde{P}(S^{U^i} = (0,2))$ and $q^{\tilde{E}A_1^{U^i}}$ are all in the range of $0, 1$, the difference between the expressions is bounded.

iv. Additional observations for a large l_2: The exact difference and ratios between the expressions in (7.18) and in (7.19) are determined by the difference between $Eq^{A_1^{U^i}}$ and $q^{\tilde{E}A_1^{U^i}}$. As explained in the analysis of the special case (Chapter 4, Subsection 4.2.3), $\lim_{i\to\infty} P(S^{U^i} = (0,2))$ depends on $\lim_{i\to\infty} Eq^{A_1^{U^i}}$. More specifically, the relation is given by:

$$
\lim_{i\to\infty} P(S^{U^i} = (1,1)) = 1 - \frac{1 - q^{A_1(0,2)+1}}{1 - q^{A_1(0,2)+1} + q^{A_1(1,1)+1}}.
$$

A similar dependency exists between the approximated values.

The accurate mean of the waiting time may, therefore, be denoted as a function of $E\left[q^{A_1(i)}\right]$. The approximated waiting time is denoted by the same function on $q^{\tilde{E}[A_1(i)]}$. There is no reason to assume that $E\left[q^{A_1(i)}\right]$ equals $q^{\tilde{E}[A_1(i)]}$, nor that they have the same limit when $i \to \infty$. We now look deeper into the two expressions.

The accurate value of $E\left[q^{A_1(i)}\right]$ is:

$$
\begin{aligned}
E\left[q^{A_1(i)}\right] &= \sum_{a=0}^{\infty} q^a P\left(A_1(i) = a)\right) \\
&= \int_{t=0}^{\infty} \sum_{a=0}^{\infty} q^a P\left(A_1(i) = a | U^(i) = t)\right) f_{U^(i)}(t) dt
\end{aligned}
$$

$$
\begin{aligned}
&= \int_{t=0}^{\infty} \sum_{a=0}^{\infty} q^{a} \frac{e^{-\lambda_1 t}(\lambda_1 t)^{a}}{a!} f_{U^{(}i)}(t)dt \\
&= \int_{t=0}^{\infty} e^{q\lambda_1 t} e^{-\lambda_1 t} f_{T(i)}(t)dt \\
&= E\left[e^{\lambda_1 U^{(}i)(q-1)}\right] \qquad (7.20)
\end{aligned}
$$

We may, therefore, conclude that:

$$
lim_{i\to\infty} E\left[q^{A_1(i)}\right] \;=\; lim_{i\to\infty} E\left[e^{\lambda_1 \cdot U^{i} \cdot (q-1)}\right] \qquad (7.21)
$$

The approximated value $q^{\tilde{E}[A_1(i)]}$ has a totally different limit:

$$
\begin{aligned}
\tilde{E}A_1^{U^i} &= \lambda_1 \cdot \tilde{E}[U^i] \\
q^{\tilde{E}A_1^{U^i}} &= q^{\lambda_1 \cdot \tilde{E}[U^i]} \\
lim_{i\to\infty} q^{\tilde{E}A_1^{U^i}} &= lim_{i\to\infty} q^{\lambda_1 \cdot \tilde{E}[U^i]} \qquad (7.22)
\end{aligned}
$$

Recall that our goal is to approximate $E\left[U^i\right]$. Therefore, we are trying to satisfy simultaneously the following two relations:

$$
\begin{aligned}
\tilde{E}[U^i] &\rightarrow E\left[U^i\right] \\
q^{\lambda_1 \cdot \tilde{E}U^i} &\rightarrow E\left[e^{\lambda_1 U^i (q-1)}\right].
\end{aligned}
$$

This, of course, does not generally apply.

In summary, we demonstrated that in some cases, for example when the number of type 1 customers waiting upon arrival is large, a good approximation is obtained when replacing the number of arrivals with its average. Under some other conditions, for example when the service rates of the two service types are similar, or when the number of type 2 waiting customers is large, the approximation is not accurate and does not have the same limit as the accurate value. Yet,an upper bound for the error of the approximation can be found. This upper bound is a function of the service rates, the arrival rate of high priority customers, and system state upon arrival.

Chapter 8

Detailed Algorithm

In this chapter we present the details of the inexplicit analytical approximation of the expected waiting time, given the system state at the time of arrival. The algorithm described can be applied to a relatively general case. As mentioned before, this work continues work began by Prof. Isaac Meilijson, from the School of Mathematics in the Faculty of Exact Sciences, Tel-Aviv University. Prof. Meilijson's work was extended to include the following:

- Servers are not necessarily statistically identical.
  - Service rates depend on the server
  - Some servers might not be able to serve certain service

  types.
- Abandonment are allowed.
- Some improvements were implemented in the algorithm itself.

The chapter is divided into three sections:

i. Model assumptions and parameters.

ii. An inexplicit analytical approximation - the basic model, no abandonment.

iii. Abandonment allowed.

We realized the approximation by a $C++$ program. The code of the program is given in Appendix A.

8.1 Model assumptions and parameters

8.1.1 Model assumptions

The service system has the following characteristics:

Several service types:

- The arrival process of each customer type is a Poisson process. Arrival rates depend on service type. Arrival processes of different service types are independent.
- Customers might abandon the system during their waiting. Customer patience is exponentially distributed. The average patience varies with service types. (see Section 8.3).

Several servers:

- The servers are independent of each other.
- Service times are exponential. Service rates depend both on service type and on server skills.
- Some servers might not be capable of serving certain service types.

A priority service discipline:

- Different priority rules may be implemented by different servers.

8.1.2 Parameters and definitions

Following is a list of the primary parameters. These parameters are used both in the approximation and later on in the simulation (see Chapter 9):

System Parameters are received as input to describe the system characteristics and the system state.

- $Pr_{[Sv\ X\ Tp]}$ is a matrix that denotes the priority level of each service type at each server. Pr(i,j) is the priority of service j at server i.
- $\mu_{[Sv\ X\ Tp]}$is the service rates matrix. $\mu(i, j)$ is the rate in which server i serves type j customers.
- $\lambda_{[Tp]}$is the arrival rates vector. $\lambda(j)$ is the arrival rate of type j customers.
- Sv is the number of servers
- Tp is the number of service types

- m is the system size: number of customers that can enter the system (used to assist in calculations and for control purposes).
- $Ty_{[m]}$is a vector that denotes the service types required by each customer in the system (ordered upon arrivals).
- $n0$ is the order number (in line) of the customer, for whom we are calculating the expected waiting time.
- $\alpha_{[Tp]}$ is the patience of customers of different service types.

Programming Parameters are additional parameters and variables, which are used in the program:

- $P_{[Sv\ X\ Tp]}$ denotes the probabilities for each server to serve each of the service types (at the current iteration). $P(i,j)$ is the probability that server i is serving a type j customer.
- $Npr_{[Sv]}$ is the number of priority levels defined to each server.
- $Ar_{[Tp]}$ is the number of arrivals that took place during the current iteration, for each service type.
- $Pes_{[Sv]}$: denotes the probability of each server to be available at the beginning of next iteration.
- $tot_{[m]}$ is the probability of each customer not to be waiting any longer (the probability that each customer has already entered service or abandoned). Interpreting the state on a fluid approximation, it describes the percentage of work that has already been performed on each customer.
- n is the number of customers that have entered the system.
- R is the system service rate. (The average iteration time is $\frac{1}{R}$).
- $Rt_{[Sv]}$ denotes the effective service rate of each server at the current iteration.
- W is the accumulated waiting time of the $n0$ customer. At the end of the process W denotes the estimated mean waiting time.

8.2 The basic model

The approximation algorithm consists of two main parts:

i. Initialization procedure which is being run only once, at the beginning.

ii. Iterative calculations that are being run over and over until our customer enters service.

We now go through the steps of each part.

8.2.1 Initialization

i. **Determining Npr vector**: first, we count the number of priority levels that each server faces. The values are not being changed later on.

ii. ***tot(i)=1* for the customers in service (first Sv customers).**

iii. $\mathbf{W} = \mathbf{0}$: we initialize the accumulated waiting time of the n0 customer to 0. W is being updated later on in every iteration.

iv. **P(i,Ty(i))=1:** at the initial stage we have the exact information regarding the service types being served by each of the servers. Therefore, we set $P(i,j) = 1$, for $j = Ty(i)$ and $P(i,j) = 0$ for $j \neq Ty(i)$.

8.2.2 Iterative calculations:

The following steps are repeated until $tot(n0) = 1$.

i. $\mathbf{R} = \sum_{\mathbf{i=1}}^{\mathbf{Sv}} \frac{\mathbf{1}}{\sum_{\mathbf{j=1}}^{\mathbf{Tp}} \frac{\mathbf{P(i,j)}}{\boldsymbol{\mu}\mathbf{(i,j)}}}$**; Calculating system service rate, R, and average iteration time**

Iteration time is a minimum of (Sv) hyper-exponential variables. We approximate it by an exponential variable, based on service probabilities of each server.

(a) $r(i) = \sum_{j=1}^{Tp} \frac{P(i,j)}{\mu(i,j)}$, for $i = 1,..,Sv$: we calculate the average time until each server completes service.

(b) $Rt(i) = \frac{1}{r(i)}$, for $i = 1,..,Sv$: we approximate the effective service rate of each server by the inverse of his mean service time in the current iteration.

(c) $R = \sum_{i=1}^{Sv} Rt(i)$: we approximate the system rate by the sum of effective service rates of all servers.

ii. $\mathbf{W} = \mathbf{W} + \frac{\mathbf{1-tot(n0)}}{\mathbf{R}}$**; updating the waiting time**

The current iteration time weighted by the probability that the n0't customer has not entered service yet, is added to the waiting time.

iii. $\mathbf{Ar(j)} = \frac{\boldsymbol{\lambda}\mathbf{(j)}}{\mathbf{R}}$, **for** $\mathbf{j = 1,..,Tp}$**; updating the system with new arrivals**

The following (iiia) to (iiif)steps are repeated for each of the service types ($j = 1,..,Tp$).

(a) $Ar(j) = \frac{\lambda(j)}{R}$: we calculate the average number of new arrivals during the current iteration.

(b) $n1 = n+\sum_{j=1}^{Tp}[Int(Ar(j))]^{+}$: $n1$ is a temporary variable that denotes the number of waiting customers including the new arrivals. If the number of new arrivals is not an integer number, we use the upper integer value.

(c) $Ty(k) = j$ for $k = n+1,..,n1$: we set the type of the new arrivals to be j.

(d) $tot(k) = 0$ for $k = n+1,..,n1$: we set the probability that each of the "just arrived" type j customers has already left queue to be 0.

(e) $tot(n1) = 1-[Ar(j)-Int(Ar(j))]$: if the number of new type j arrivals is not an integer, then the probability of the last customer to be waiting is set to the appropriate fraction. For example, if $Ar(j) = 2.1$, then for two type j customers we set $tot(k) = 0$, and for the third type j customer we set $tot(i) = 0.9$.

(f) $n = n1$: we add the new arrivals to the actual queue.

iv. $\mathbf{Pes(i)} = \sum_{\mathbf{j=1}}^{\mathbf{Tp}} \mathbf{P(i,k)} \cdot \frac{\mu\mathbf{(i,j)}}{\mathbf{R-Rt(i)}+\mu\mathbf{(i,j)}}$, **for** $\mathbf{i = 1,..,Sv}$: **calculating service completions probabilities**

$Pes(i)$ is the probability of each server to be the one who completes service at the next service completion event, or in other words the working capacity that could be assigned to server i at the end of the iteration.

The following is repeated for each of the servers ($i = 1,..Sv$):

(a) $Rt(i)$ is the current service rate of server i (as was found in step ib). Given that server i is serving a type k customer, his current service rate is $\mu(i,k)$. The entire system's service rate, given that server i is serving a type k customer, is: $R - Rt(i) + \mu(i,k)$. Since we use exponential approximation for the service times, the probability that server i will be the one who completes service in the next service-completion event is approximated by: $[\frac{\mu(i,k)}{R-Rt[i]+\mu(i,k)}]$

(b) $Pes(i) = \sum_{j=1}^{Tp} P(i,k) \cdot \frac{\mu(i,j)}{R-Rt(i)+\mu(i,j)}$: we use the law of total probability to update $Pes(i)$.

v. **Assigning tasks for next iteration and updating service configuration probabilities** $\mathbf{P(i,j)}$

In the last part we assign to the servers new tasks to work on in the next iteration.

We begin with server $i = 1$, and perform the following steps:

(a) We look for the next customer in line according to the priority rules implemented by server i (the longest waiting customer among the highest priority waiting customers). (First, we check if there are waiting customers with highest priority level, then we proceed to second priority level etc...). Let k be the index of this customer.

(b) If the fraction of work required by customer k (the probability that customer k has not entered service yet), $1 - tot(k)$, is smaller than server's i capacity (the probability that server i will be available, $Pes(i)$), then we assign the work required by customer k to server i.

We update the system as follows:

- $Pes(i) = Pes(i) - (1 - tot(k))$: server's i capacity is reduced.
- $P(i, Ty(k)) = P(i, Ty(k)) + (1 - tot(k))$: the probability that server i is serving a Ty(k) customer is increased.
- $tot(k) = 1$: the probability that customer k is still waiting and the amount of work that he requires are set to 0.
- We go back to step va in order to identify the next customer in queue (for server i) and to assign additional tasks to server i.

(c) If the fraction of work required by customer k is larger than server i's capacity, then we assign the maximal amount possible from the work required by customer k to server i. We update the system as follows:

- $tot(k) = tot(k) + Pes(i)$: the work left for customer k (or the probability that he is still waiting) is reduced.
- $P(i, Ty(k)) = P(i, Ty(k)) + Pes(i)$: the probability that server i is serving a $Ty(k)$ customer is increased.
- $Pes(i) = 0$: the availability of server i is set to 0.
- We set $i = i + 1$, and go back to step va to assign tasks to the next server.

vi. **Stopping condition:**

(a) If the probability that our customer has entered service is still smaller than 1 ($tot(n0) < 1$), then:

- We check that the system is not overloaded (if $n >= m$ we stop the process and give a notice).
- We go back to step i and generate another iteration.

(b) If $tot(n0) = 1$ then the process is over and the estimated waiting time is W.

8.3 Abandonment allowed

To allow abandonment, a new routine was added to the algorithm between step (ii) in which waiting time is updated and step (iii) in which new arrival are considered.

By this routine, we calculate for each customer in line the probability to abandon during the current iteration. The time until an abandonment of a type(j) customer is exponential with a rate $(\alpha(j))$. Iteration time is approximated by an exponential with a rate R. The probability that a specific type j waiting customer will abandon the line before the iteration is completed, is approximated by $\frac{\alpha(j)}{\alpha(j)+R}$. This probability is multiplied by the probability that the customer is still waiting at the beginning of the iteration: $(1 - tot(k))$, for the k'th customer. Next, the probability of the customer to be waiting is being updated. The probability that the k'th customer will still be waiting at the end of the current iteration is, hence, updated as follows, for each customer in queue $(k = Sv + 1, .., n)$:

- If $((tot(k) = 1)$ or $(k = n0))$, then we move on to the next customer in line.

- $tot(k) = tot(k) + (1 - tot(k)) \cdot \frac{\alpha(Ty(k))}{\alpha(Ty(k))+R}$.

In Chapter 9, it will be shown that when abandonment is allowed and the above procedure is used, the accuracy of our approximation decreases.

Chapter 9

Simulation

As mentioned before, one of the options for estimating waiting times in complex systems is running a simulation. A simulation may be applied to complicated systems with many parameters such as: a large number of servers with different skills, different priority policies for different servers, abandonment etc. Simulations can often be used for a large modelling scope, but computation times might be long.

We used simulation for two purposes:

i. Compare the mean waiting time as estimated by our approximation to that estimated by a simulation, in order to check the accuracy of our approximation.

ii. Confirm the existence of a simple approximation of the mean waiting time in large systems. Intuitively, when an already large system is enlarged by a certain factor, waiting times should not be significantly affected. To check this assumption, we examine the mean waiting time as a function of the queue size and the number of servers. As one can see in Section 9.2, it seems that a simple relation between number of servers, queue size, and the expected waiting time can be formulated for large systems.

For simplification and consistency reasons and in order to be able to compare the approximation performance in different models, some common characteristics are kept fixed in all the examples:

- Two service types.
- Service type 1 has a static non-preemptive priority over service type 2.
- All servers are statistically identical.
- Upon arrival, all the customers being served are type 1 customers.

Other parameters are changed according to the issue investigated.

A salient finding is that when abandonment is assumed out, our approximation yields highly accurate results. As expected, waiting time estimations for high priority customers were extremely accurate. Waiting time estimations for low priority customers were also very accurate in many cases, for example when the same load was created by each of the service types. Usually, the results obtained by the approximation were pretty close to simulation's ones, and almost no major errors were observed. Deviation from simulation results was below 5% in most of the cases. The accuracy of our approximation increases with the number of servers and with queue size, but decreases with the offered load of high priority customers ($\frac{\lambda_1}{Sv\cdot\mu_1}$), probably due to the greater effect of new arrivals. We have not found any relation between service rates and accuracy of the approximation. Yet, further investigation is needed in order to conclude that such a relation does not exist.

When abandonment are allowed, the accuracy of the approximation significantly decreases. With abandonment, the approximation overestimated the mean waiting time, with deviations from simulation's results typically being in the range of 11%-15%. Further research is required both in order to understand the reasons for this and to possibly improve the abandonment procedure.

In this chapter we discuss our conclusions and present some numerical results to support and explain those conclusions. The detailed results of all experiments can be found in Appendix C. The code of the simulation program in C++ is given in Appendix B.

9.1 Comparison of the approximation with simulation

In this section we present numerical comparisons of the mean waiting time, as estimated by our approximation and by simulations.

We compare the results and discuss the main findings, focusing on several aspects:

i. Waiting times of high priority customers

ii. Balanced system

iii. Number of servers

iv. Queue size

v. Service rates

vi. Abandonment

vii. Calculation times

Before getting into the numerical results, we would like to make several observations regarding the nature of comparison:

- Most comparisons relate to the waiting time of low priority (type 2) customers. Estimation of waiting times of low priority customers is more challenging than that of high priority ones. The accuracy of our approximation is expected to be lower for low priority customers than for high priority ones, and therefore this case is more interesting. However, we make some experiments regarding high priority customers. The results obtained by our approximation in these experiments are highly accurate, as one can see in Section 9.1.1.

- As already explained is Subsection 6.3.4, our approximation might loose accuracy as the arrival rate of type 1 customers increases (and their contribution to the system load increases). We start by showing that when the load generated by the two types of service is equal, the results of our approximation almost equal those of a simulation. Our goal is to investigate the sensitivity of our approximation to different parameters, such as service rates and number of servers. This can not be done when all the results are highly accurate. Concluding regarding the behavior of the approximation under different circumstances is easier in extreme cases. We, therefore, continue with systems in which type 1 customers alone create a load of $\frac{\lambda_1}{Sv \cdot \mu_1} = 0.9$. We select to use the extreme condition in order to analyze a "worst case" scenario. We should keep in mind that the results for more balanced, and in a way more realistic, systems are expected to be more accurate.

- One should recall that simulation might have small inaccuracies as well (resulting, for example, from too little runs or from the way random variables are generated).

- The algorithm and simulation were run on a typical home-PC. Machine resources were sometimes shared with other processes. Therefore, any reference to running times should be reviewed not in absolute terms but with respect to the difference between the approximation run time and the simulation run time.

9.1.1 Waiting times of high priority customers

Estimations of waiting time of a high priority customer (type 1) who, upon arrival, finds only type 1 customers in service is fairly easy. Estimations by our approximation are expected to be accurate.

However, we ran one example for a system with two servers. The parameters that were used are: $\mu_1 = 15$, $\mu_2 = 30$, $\lambda_1 = 25.5$.

The approximation results in this case were accurate, just as expected. We have run the model for 25 values of L^0 while ($L_1^0 = 0,\ 5,\ 10,\ 15,\ 20$) and ($L_2^0 = 0,\ 5,\ 10,\ 15,\ 20$). For all those cases the mean waiting time estimated by the approximation was equal to the waiting time estimated by a simulation, except for two cases in which there was an error of 1%.

9.1.2 Results for a balanced system

We define the load created by type 1 customers by $\frac{\lambda_1}{Sv \cdot \mu_1}$. The total load in the system (in steady state) is determined by $\frac{\lambda_1}{Sv \cdot \mu_1} + \frac{\lambda_2}{Sv \cdot \mu_2}$, and must be smaller than 1 when abandonment does not exist. The waiting time of a specific type 2 customer (given the system state upon arrival), is affected by the load created by type 1 customers but not by the total load. (The service rate of type 2 customers affects the waiting time, but the arrival rate does not). We estimated the mean waiting time of a type 2 customer in two cases:

i. Balanced system - The load created by each type of customers is the same. The total load of the system is 0.9 and the load created by each type of customer is 0.45. This model may describe a system in which type 1 customers receive higher priority due to marketing and management considerations only.

ii. Unbalanced system - The load created by customers with a low priority is very small. The load created by customers with high priority alone is high (0.9). That may describe, for example, a call center in which the major activity is of type 1, and demand for type 2 services is rare.

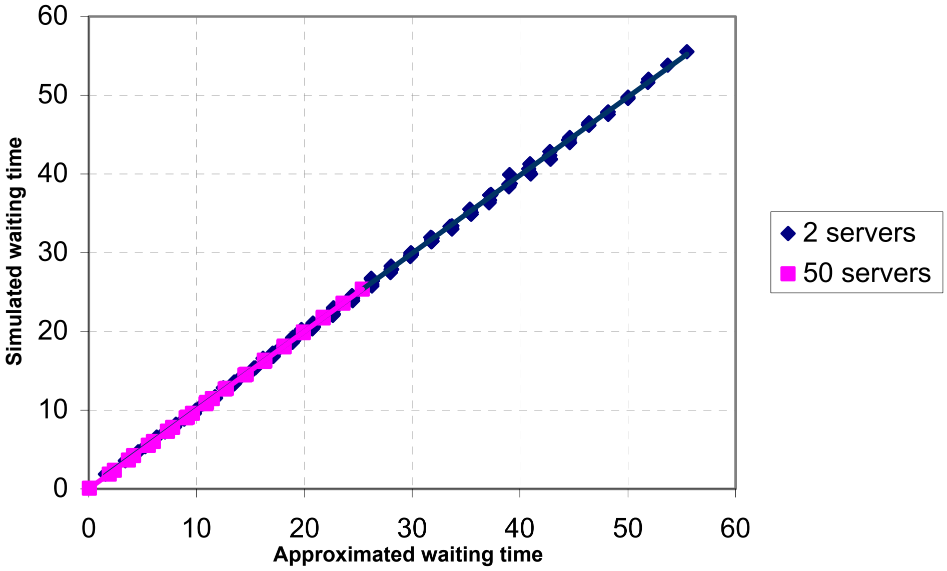
To evaluate the accuracy of our approximation in balanced systems, we estimated the mean waiting time of a type 2 customer in a system with 2 servers and in a system with 50 servers. In both systems, the load created by each type of customers was 0.45 (and the total load 0.9). We used both our approximation and a simulation to estimate the waiting times for each of the systems, and then compared the results. Service rates were the same in both cases: $\mu_1 = 30$ and $\mu_2 = 15$. Arrival rates were set to satisfy the required load. In the case of two servers we used $\lambda_1 = 27$, $\lambda_2 = 13.5$ (note, again, that λ_2 does not affect the results). In the case of 50 servers we used $\lambda_1 = 675$ and $\lambda_2 = 337.5$.

As illustrated in Figure 9.1, the estimations obtained by our approximation are almost completely accurate when the system is balanced.

For a system with 2 servers, we ran the program for about 120 different values of (L_1^0, L_2^0). The number of customers waiting upon arrival varied between 0 to 10 for each type of customers. The largest error in the model of 2 servers was 17%. That error occurred for the case of no customers in line upon arrival ($L^0 = (0, 0)$). For all other values of (L_1^0, L_2^0) the gaps between the results of our approximation and those of a simulation were up to 5%. In most cases, the error was not above 2%.

For a system with 50 servers the differences between the results obtained by our approximation and those obtained by a simulation were even smaller. The estimated waiting times obtained by our approximation were almost in all cases equal to the values obtained by a simulation. We compared the results for 36 values of (L_1^0, L_2^0), changing the initial queue size in the range of $(0, 25, 50, ..., 125)$ for each type of customers.

Figure 9.1: simulation vs. approximation in a balanced system $\frac{\lambda_1}{Sv \cdot \mu_1} = \frac{\lambda_2}{Sv \cdot \mu_2} = 0.45$



As expected, the difference between approximation results and simulation results in the case of an unbalanced system is much bigger. The results are discussed in the next sub-sections, and detailed tables are presented in Appendix C.

All the results presented in the next sections refer to an unbalanced system, where the load created by type 1 customers is high (usually 0.9). As we said before, we believe that it represents a ”worst case scenario”, and therefore if the accuracy of our approximation will be acceptable in this case, it will also be acceptable under more moderate and common conditions. Analyzing an extreme case also makes it easier to identify trends and relations. (It is difficult to identify the affect of different factors on the accuracy of our approximation, in a model that always yields high accuracy).

9.1.3 Number of servers

We have estimated the mean waiting time of a type 2 customer for several systems, with the following characteristics:

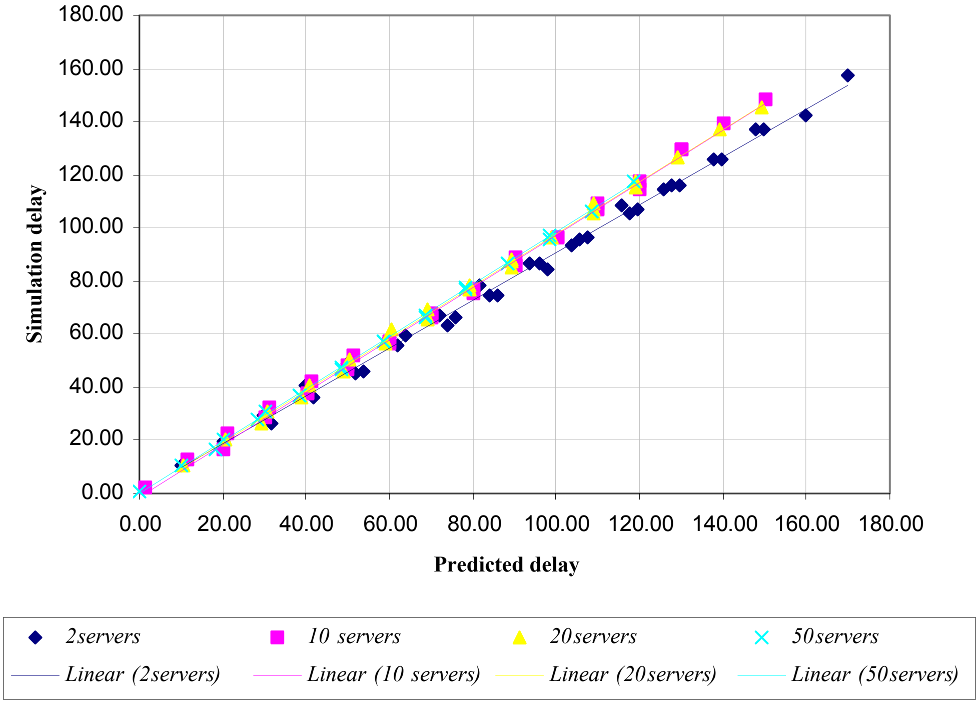
- Service rates are the same in all systems and are set to $\mu_1 = 30$, $\mu_2 = 15$.
- Number of servers varies: systems with 2 servers (very small), 10 servers (small), 20 servers (small-medium) and 50 servers (medium) were analyzed.
- Arrival rate of type 1 customer was set to satisfy a desired load factor, so that $\frac{\lambda_1}{Sv \cdot \mu_1} = 0.9$. Arrival rate of type 2 customers has no affect on the results.
- No abandonment.
- Queue size vastly varies.

Estimations obtained by a simulation vs. those obtained by our approximation for each of the systems are presented in Figure 9.2. In addition to the results themselves we drew the linear trends for each of the systems.

A salient finding from the comparison is that as the number of servers increases, the approximated waiting times get closer to the simulated ones. This can be concluded from the rates of the linear trend-lines. The rate of the linear trend line, approaches the value of 1 as the number of server increases. Also the lowest trend-line is that of the system with two servers. (Since all rates are smaller than 1, a higher trend-line indicates higher accuracy of our approximation). The lines of a system with 10 servers and that of a system with 20 servers are close to each other. The highest trend-line is that of a system with 50 servers. That finding is clearer for high waiting times than for short ones.

We also compared the accuracy of different systems (different number of servers), when the entire system grows by a certain factor. To do that the queue size of type i customers (i=1,2) is denoted by $(k_i \cdot Sv)$ rather than by absolute values. We compare the accuracy of the approximation for different number of servers (Sv). To measure the accuracy of the approximation we use the value obtained when dividing the approximated mean waiting time by the simulated mean waiting time. The approximation accuracy is determined by the distance of that value from 1. Figures 9.3 and 9.4 show the accuracy of the approximation for systems of different sizes ($Sv = 2,\ 10,\ 20,\ 50$). In each figure k_1 is constant and the results are presented as a function of k_2. We present the results for two cases: when the number

Figure 9.2: Simulation vs. approximation for different numbers of servers



of type 1 customers waiting upon arrival equals the number of servers ($K_1 = 1$, see Figure 9.3), and when the number of type 1 customers waiting upon arrival is 1.5 times the number of servers ($K_1 = 1.5$, see Figure 9.4).

As one can see, usually, for a given ratio of queue size to number of servers (given k_1, k_2), approximation accuracy is higher as the number of servers increases. Though few results do not support this conclusion, it describes the general behavior. This is clearly observed as queue size increases. Another interesting finding is that deviation range is larger for smaller systems. In other words, the differences in the approximation accuracy for different values of k_1, k_2 are bigger in small systems. For example, as can be seen in Figure 9.3: the accuracy levels for a system with 2 servers vary between 1.15 (for $k_2 = 3.5$) to 0.99 (for $k_2 = 0$), while accuracy levels for a system with 50 servers vary only between 0.99 (for $k_2 = 0$) to 1.03 (for $k_2 = 2$). The above trends are clearly illustrated by the line displaying results for a system with 50 servers (in both figures). It seems that this line is almost stabilized very close to the value of 1.

Figure 9.3: Comparison when queue size of type 1 customers equals the number of servers

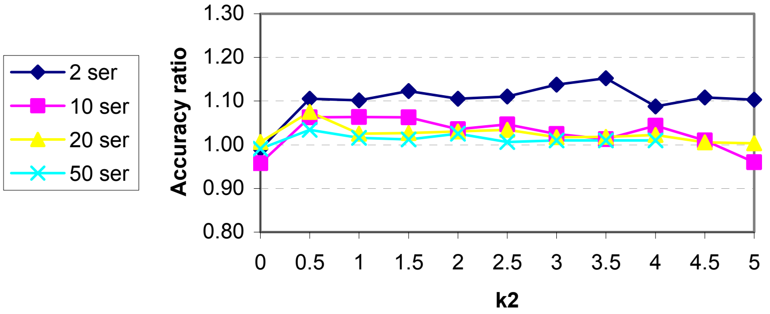
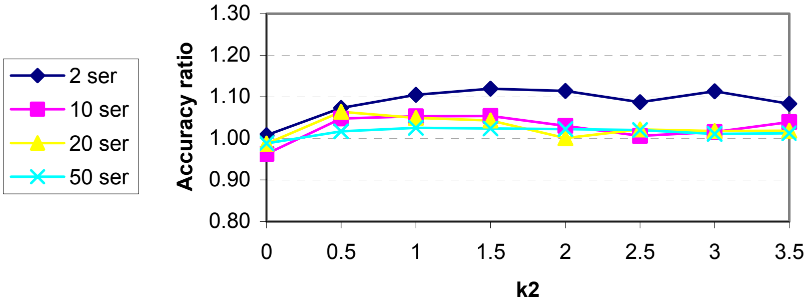


Figure 9.4: Comparison when queue size of type 1 customers is 1.5 times the number of servers



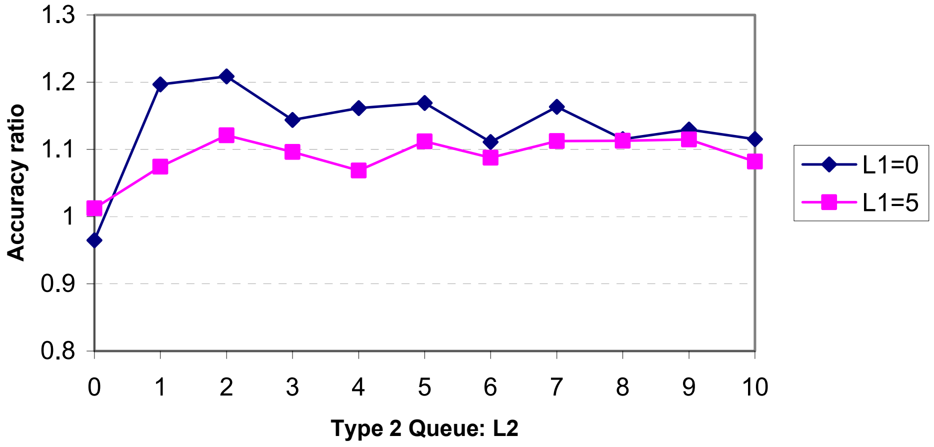
9.1.4 Queue size

We used the results of the experiments for unbalanced systems, in order to evaluate the accuracy of our approximation as a function of L^0, when the number of servers remains fixed.

Generally speaking, a large deviation of the approximated value from the simulated one appears only when queue size is small. For example, in the system with 10 servers when $L^0 = (0,0)$ the approximated mean of waiting time is 1.28 minutes, while the simulated value is 2.25 minutes (error of 43%). For $L^0 = (0,5)$ the approximated mean of waiting time is 20.8 minutes and the simulated value is 16.8 (an error of 24%). For all other values of queue size the deviation is in the range of ±6%.

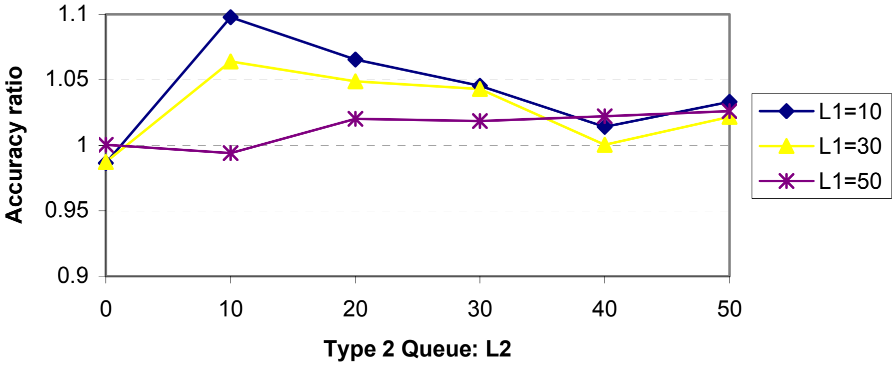
The accuracy of the approximation improves both when L_1^0 or L_2^0 increases. Yet, we identify that the increase of L_1^0 has a faster impact. For example in a system with 10 servers, while the approximated value for $L^0 = (0, 10)$ is 1.06 times the value received by a simulation, the ratio for $L^0 = (10, 0)$ is only 0.96 (that means a deviation of 4% as opposed to 6%).
In a system with 2 servers we saw a similar trend. In Figure 9.5 we present the accuracy of our approximation compared to simulation as a function of the number of type 2 waiting customers. This is done for two fixed values of L_1: $L_1^0 = 0$, meaning no type 1 customers waiting, upon arrival, and $L_1^0 = 5$. We found that as the number of type 2 waiting customers increases, the accuracy of our approximation increases as well. In addition, for a given number of type 2 waiting customers, the accuracy of our approximation usually increases with the number of type 1 waiting customers.

Figure 9.5: Queue size effect on approximation accuracy, 2 servers



A similar though a less clear trend is observed for a system with 20 servers, as illustrated in Figure 9.6. For larger systems, as accuracy levels get higher in general, this trend is not observed any longer.

Figure 9.6: Queue size effect on approximation accuracy, 20 servers



9.1.5 Service rates

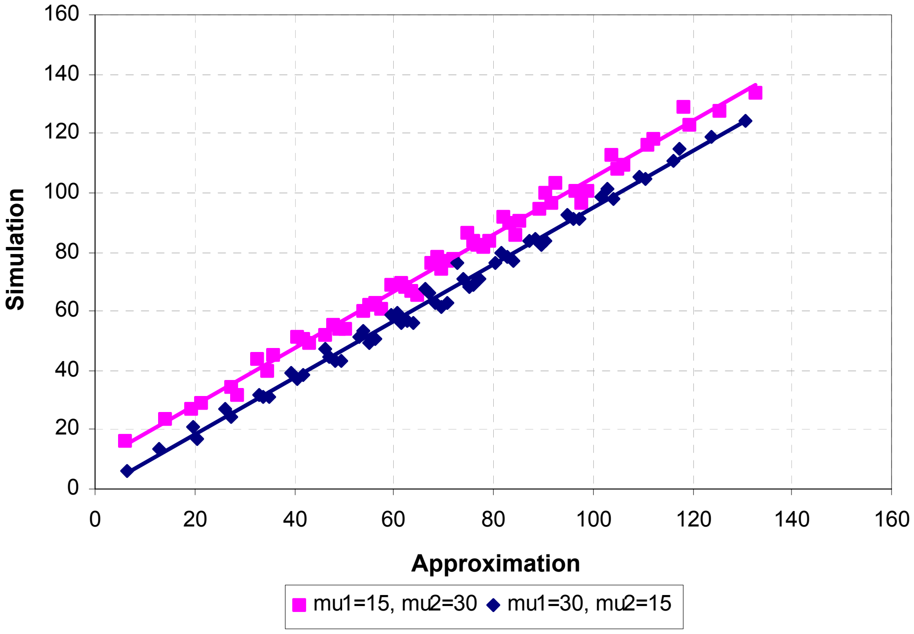
To evaluate the impact of service rates on the accuracy of our approximation, we have compared the results in two similar systems with two servers. In the first system the service rate of type 1 customers was larger than that of type 2 customers ($\mu_1 = 30$, $\mu_2 = 15$), while in the other system service rates were flipped ($\mu_1 = 15$, $\mu_2 = 30$). The arrival rates were changed with the service rates, keeping a ratio of $\frac{\lambda_1}{Sv \cdot \mu_1} = 0.85$. As can be seen in Figure 9.7, the accuracy of our approximation was similar in the two cases.

However, this analysis is not enough to determine that the service rates do not affect the approximation accuracy. More research is required in order to understand the relation between service rates and approximation accuracy. When referring to our results one should keep in mind that only two sets of service rates were examined. The fact that the service rates in the two sets were simply flipped could also impact the result.

9.1.6 Abandonment allowed

We examined the differences between the mean waiting time of a type 2 customer as obtained by a simulation and by our approximation in a model with abandonment

Figure 9.7: Service rates effect on approximation accuracy, 2 servers

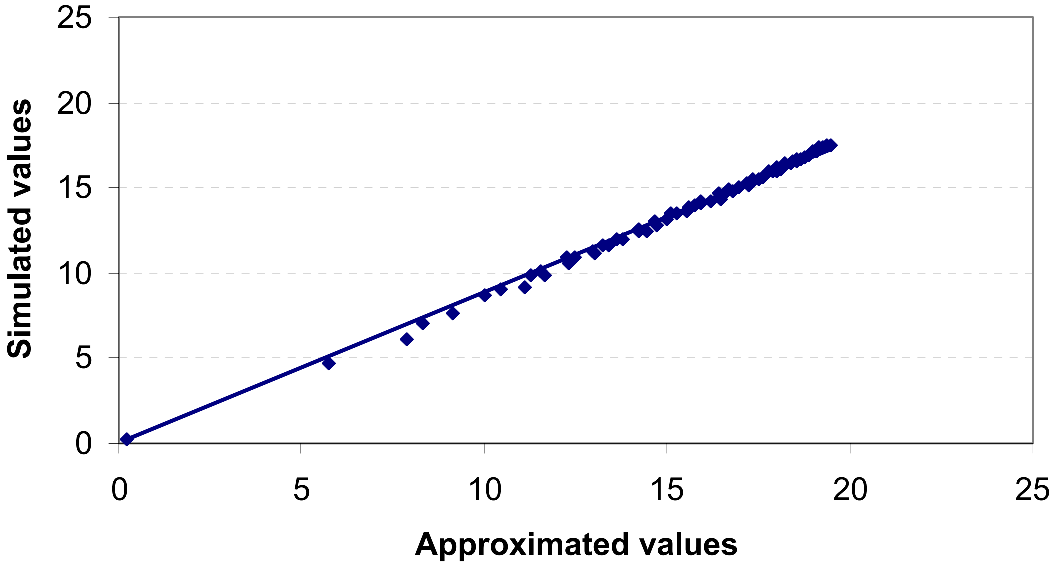


and with the following characteristics:

- 50 servers
- Service rates are $\mu_1 = 30$, $\mu_2 = 15$
- Arrival rate of type 1 customers is $\lambda_1 = 1350$ (satisfies a ratio of $\frac{\lambda_1}{\lambda_1 \cdot Sv} = 0.9$).
- The patience of customers is exponential with parameter $\alpha = 12$ both for type 1 and type 2 customers. That means that the average time customers are willing to wait for service is 5 minutes.
- Number of type 1 customers waiting upon arrival receives values of $L_1^0 = (0, 25, 50, ..., 125)$. Number of type 2 customers waiting upon arrival receives values of $L_2^0 = (0, 25, 50, ..., 250)$

As illustrated in Figure 9.8, almost in all runs the approximation overestimated the mean of the waiting time. The deviation of the approximated value from the simulated value was larger than for models without abandonment. For most cases the deviation was between 11% and 15%. Accuracy of our approximation improves as the original queue size increases, but improvement rate is lower than the one observed in models without abandonment.

Figure 9.8: Abandonment is allowed, 50 servers



9.1.7 Calculation times

While computation times using simulations become very long as system size increases, computation times using the approximation remain significantly shorter. For example:

- For a system with 50 servers (unbalanced) obtaining an estimation by a simulation might take more than 10 hours (!), while estimating the waiting time using our approximation takes up to 20 minutes. When running the balanced system, there are less arrivals of type 1 customers and therefore fewer iterations. Running a simulation then takes about 20 minutes, and running our approximation takes only a few seconds.

- For a system with 10 servers simulation time might take more than 30 minutes, approximation time is not longer than 15 seconds.
- For a system with two servers approximation results are immediately obtained (less than 0.1 seconds), while simulation can take more than 10 minutes.

To obtain each result by a simulation, we ran 2000 iterations, and averaged over them. Obviously, if the number of iterations is smaller, computation times will be shorter.

It should be mentioned, again, that programs were run over a typical home-pc. Machine resources were sometimes shared with other processes. Hence, times should not be referred to in absolute terms, but in respect to differences between the run-time of our approximation and the run-time of a simulation.

9.2 Waiting time as a function of queue size and number of servers

Simple approximations for attributes of large systems can sometimes be obtained using fluid or diffusion approximations. The existence of such an approximation for the anticipated waiting time enables easy and immediate estimations for large systems.

Intuition suggests that when an already large system is enlarged by a certain factor, waiting times should not be significantly affected. In other words, if servers are added to the system, but arrival rates and queue size upon arrival grow by the same ratio, waiting time should stay more or less the same. Motivated by this hypothesis, we used simulation results to examine the waiting time of a type 2 customer, as a function of the number of servers and to analyze its limit, when the number of servers increases (and all other parameters respectively increase as well). To do so, we denote the number of waiting customers as a function of number of servers. As we have done before, we define k_1, k_2 so that: $L_1^0 = k_1 \cdot Sv$ and $L_2^0 = k_2 \cdot Sv$. When the number of servers is very large, we approximate the system by an $M/G/1$ system (with the same priority policy). The service rate is approximated by the service rate of a single server multiplied by the number of servers. We can, therefore, estimate the mean waiting time of a type 2 customer (given that all customers served upon arrival are type 1 customers, as explained at the beginning of this chapter) by:

$$\begin{aligned} & E[W_2(S^0 = (Sv, 0); L^0 = (k_1 \cdot Sv, k_2 \cdot Sv))] \\ \simeq \; & \left[\frac{1}{Sv \cdot \mu_1} + \frac{k_1 \cdot Sv}{Sv \cdot \mu_1} + \frac{k_2 \cdot Sv}{Sv \cdot \mu_2}\right] \cdot \frac{Sv \cdot \mu_1}{Sv \cdot \mu_1 - Sv \cdot \lambda_1}. \end{aligned} \tag{9.1}$$

When the number of servers is large enough, we get the following limit:

$$\begin{aligned}&\lim_{Sv\to\infty} E[W_2(S^0=(Sv,0);L^0=(k_1\cdot Sv,k_2\cdot Sv))]\\&=\left[\frac{k_1}{\mu_1}+\frac{k_2}{\mu_2}\right]\cdot\frac{\mu_1}{\mu_1-\lambda_1}.\end{aligned} \tag{9.2}$$

The result in (9.2) implies that the average waiting time in large systems is not affected by the number of servers, but only by the ratio between the queue size of each service type and the number of servers. If we increase the number of servers in our model, and proportionally increase queue size and arrival rates, we expect to see waiting times converging towards the limit in (9.2).

We looked into estimations of average waiting times as a function of the number of servers and of the ratios k_1 and k_2. Some examples are presented in Table 9.1. (The results are for the model with $\mu_1=30$, $\mu_2=15$, $\lambda_1=27\cdot Sv$ and no abandonment).

As one can see from the results, the simulated mean of waiting time begins to stabilize as the number of servers increases. The simulated values for systems with 10, 20 and 50 servers are usually similar. Yet those values are close but not equal to the theoretical limits. That either means that the systems are still too small, or indicates regular simulation errors.

Table 9.1: Mean waiting time as a function of proportions between queue size and number of servers

Number of servers	k_1	k_2	*Simulation (minutes)*	*Theoretical limit (minutes)*
2	0.5	0.5	36.0	30
10	0.5	0.5	28.7	30
20	0.5	0.5	26.5	30
50	0.5	0.5	27.7	30
2	0.5	2.5	116.1	110
10	0.5	2.5	107.2	110
20	0.5	2.5	105.6	110
50	0.5	2.5	106.3	110
2	1	1	63.5	60
10	1	1	56.5	60
20	1	1	57.6	60
50	1	1	57.5	60
2	1	1.5	86.6	80
10	1	1.5	75.4	80
20	1	1.5	77.0	80
50	1	1.5	77.5	80
2	1	2.5	126.1	120
10	1	2.5	114.8	120
20	1	2.5	115.1	120
50	1	2.5	117.7	120
2	1	0.5	45.2	40
10	1	0.5	37.7	40
20	1	0.5	36.3	40
50	1	0.5	37.2	40
2	1.5	1	74.9	70
10	1.5	1	66.6	70
20	1.5	1	65.8	70
50	1.5	1	66.7	70
2	1.5	1.5	96.0	90
10	1.5	1.5	85.5	90
20	1.5	1.5	85.4	90
50	1.5	1.5	86.3	90
2	1.5	2	115.7	110
10	1.5	2	106.9	110
20	1.5	2	109.0	110
50	1.5	2	106.1	110

Chapter 10

Summary

In this work we have analyzed different models of call-centers. For each model we developed methods for estimating waiting times, based on the system state at the time of estimation. Such estimations can be used to inform customers about their anticipated waiting times, and thus to contribute both to customer satisfaction and to the objective performance of the system.

We started by estimating waiting times in systems with a FCFS service. Then, motivated by the structure of modern call centers, we proceeded to service models of skills-based-routing and priorities. We focused on models with static priority, and examined several alternatives. First we demonstrated exact calculations using difference equations and matrix geometric solutions. Then we suggested an approximation for the mean of the anticipated waiting time. The approximation can be applied to a quiet wide range of models. Simulations can also be used, and we have, indeed, numerically compared the results of our approximation to these of a simulation for different systems. Doing that, we found that the approximation is pretty good for many cases, and identified some of its weaknesses. For example, the accuracy of our approximation increases with system size, but decreases when abandonment is allowed.

10.1 Future research

Much is left to do in order to have an accurate and practical method for estimating waiting times. One may continue our work in any of the following directions:

i. Improve and elaborate on our approximation

ii. Analyze additional models

iii. Verify estimations by empirical data

iv. Develop novel approaches, for example based on fluid and diffusion approximations

We will now elaborate on the suggested directions.

10.1.1 Improve and elaborate on our approximation

Both the approximation method and its realization (the C++ program in Appendix A), can be improved in terms of accuracy and ease of use.

- As the approximation is implemented today, feeding the input might not be very convenient for large systems. For example, the characteristics of each and every server is to be specified: service rates, priorities and initial service configuration. It is reasonable to assume that when the number of servers is large, servers may be divided into groups, so that all the servers in the same group are statistically identical. Each group will be characterized, and the number of servers belonging to it will be specified. This compact form of system description includes all the required information. As explained in Chapter 8, doing that will also enable to replace parts of the approximation with more accurate procedures (for assigning tasks and for calculating average iteration times).

- Similarly to the situation with servers in the current implementation, each and every customer in the system should be characterized separately. It is also possible to refer to groups of customers rather than to each customer separately. The number of customers of each type will be maintained and updated. This will significantly ease the input process, and may also lead to more efficient computations (in terms of computation time).

- Currently, we approximate the mean of the waiting time. However, more is often required. We recommend to refine the approximation, so that it will include more information about the anticipated waiting. One possibility, which we believe will be fairly easy to implement, is an estimation of the standard deviation of the waiting time.

- Obviously, some inaccuracy occurs when approximating the time of each iteration (time between successive completions of service). Numerical comparisons of our approximation to simulation's results leads us to the conclusion that we overestimate iteration times. Using alternative methods to approximate the hyper-exponential iteration times can possibly refine the algorithm and improve its accuracy. One example for such an alternative is to approximate the mean time of each iteration by an exponential with a point mass at the origin. This method can be used to approximate random variables with a standard deviation higher than their average, which is the case for hyper-exponentials.

- Numerical comparisons to simulation results showed that the accuracy of our approximation decreases when abandonment is allowed. As abandonment can not be neglected in telephone systems, further study is recommended here. We would like to mention that most comparisons were made for systems without abandonment. Therefore, our understanding of the reasons of the decrease in accuracy and of the conditions under which it occurs, might not be sufficient yet.

10.1.2 Analyze additional models

The description of call centers as queueing systems may often go beyond the models covered in this work. Developing methods for estimating waiting times for as wide range of call center schemes as possible seems to be interesting and useful. The following are only a few examples of assumptions that can be removed or changed regarding the operational characteristics of the call center:

- A non-homogenous arrival process
- A non-Poisson arrival process
- Non-exponential service times
- Non-exponential patience (as indicated in [35] to be the case in a real call center).

Different assumptions regarding the design of services and the service policy can also be made. Some examples include:

- Contact centers: modern call centers provide services via e-mail, internet, and fax, in addition to telephone services. The different service channels differ by their requirements as well as by their sensitivity to attributes, such as waiting times, service times, and preemption. For example, back-office work (answering a fax or an e-mail) can be stopped upon receiving a telephone or an on-line call.

- Non-static priorities: while with a static-priority, the assignment rules do not change with queue-size or with waiting durations, it is possible to follow pre-defined assignment rules that depend on the system state. For example, type 1 customers get a priority over type 2 customers, unless the number of type 2 waiting customers exceeds a certain value (absolute or relative). Non-static priority policies are sometimes more efficient than static priority ones.

10.1.3 Verify estimations with empirical data

Though we have compared the results of our approximation to these of a simulation, there is no substitute for empirical data. The most reliable way to evaluate the accuracy of each estimation method is to verify it against empirical data. Almost all call centers today utilize advanced systems for data collection. It should be possible to extract, with a reasonable effort, the system state at the time of each individual call and the waiting time of that call. The waiting of each call, given the system state at the time of the call, can be estimated using the evaluated estimation method. Then the results should be compared against the measured waiting times.

10.1.4 Develop novel approaches

Simple approximations can often be obtained for large systems operated in heavy traffic. The analysis of large systems is done using special methods, such as fluid and diffusion approximations.

Though in this work we have not focused on large systems, we touched some related issues. For example, in 9.2, we suggested that the average waiting time in large systems is not affected by the number of servers, but only by the ratio between the queue size of each service type and the number of servers. This was supported by simulation results.

Another interesting approach, that should be further developed is estimations of waiting times by the waiting time of the longest-waiting-customer.

We believe that there is much to do with respect to analysis of large systems. Developing novel approaches can lead to new and simple methods for estimating waiting times.

Appendix A

Approximation - C++ Program

In this appendix we present the C++ program for the simulation.

```
// algorithms.Cpp
//
#include "algorithms.h"

using namespace std;

////////////////////////////////////////////////////////////////////////////
//class CInput
////////////////////////////////////////////////////////////////////////////
CInput::CInput()
{
    Size =  n0 =  Ser  = Typc = Type_n0 = -1;
}
//---------------------------------------------------------------------------
CInput::CInput(const CInput& r)
    :Size(r.Size),
    Ser(r.Ser),
    n0(r.n0),
    Typc(r.Typc),
    Type_n0(r.Type_n0),
    Pr(r.Pr),
    Mu(r.Mu),
    La(r.La),
    Alpha(r.Alpha),
    Type(r.Type),
    Queue(r.Queue)
{
}
//-----------------------------------------------------------------------------
void CInput::ReadInput(istream& is) {
    //Read line 1 - numbers of: servers, service types, system size, and Type_n0
    //(changed by Efrat on 30/08/00).

    is >> Ser >> Typc >> Size >> Type_n0 ;

    if(!is || (is.get() != '\n'))
        ExitError("Error in Input Parameters");

    int n = -1;

    //Read "Ser" number of lines: each line
    // contains number of "service types" ints - for "Pr"
    Pr.resize(Ser,IntVect(Typc,0));
    if(-1 != (n = iStream2IntMtx(Pr,is)))
    {
        cout << endl << "Error in Pr Matrix, line: " << n << endl;
        exit(0);
    }

    //Read "Ser" number of lines: each line
    // contains number of "service types" doubles - for "Mu"
    Mu.resize(Ser,DblVect(Typc,0));
    if(-1 != (n = iStream2DblMtx(Mu,is)))
    {
        cout << endl << "Error in Mu Matrix, line: " << n << endl;
        exit(0);
    }

    //line  - number of "service types" doubles - for "La"
    La.resize(Typc);
    if(!iStream2DblVect(La,is))
        ExitError("Error in La");

    //line  - number of "service types" doubles - for "Alpha"
    Alpha.resize(Typc);
    if(!iStream2DblVect(Alpha,is))
        ExitError("Error in Alpha");
    //line  - number of "ser" ints - for "Type"
    Type.resize(Ser);
    if(!iStream2IntVect(Type,is))
        ExitError("Error in Type");

    //line  - number of "Typc" ints - for "Queue"
    Queue.resize(Typc);
    if(!iStream2IntVect(Queue,is))
        ExitError("Error in Queue");
}
//------------------------------------------------------------------------------
void CInput::Prepare4Calc() {

    //Initializing the queue (assigining types and determining n0)

    Type.resize (Size);
    n0 = Ser;
    for(int j =0; j < Typc; j++)
    {
        for (int k=n0; k < n0 + Queue[j]; k++)
        {
            Type[k] = j;
        }
        n0 += Queue[j];
    }
    // setting type and place of the customer
    // for whom we are calculating waiting time

    Type[n0] = Type_n0;
}
//////////////////////////////////////////////////////////////////////////////
//class CAnalityc
//////////////////////////////////////////////////////////////////////////////

// Travers procedure: a recursive procedure for accurate
//computation of the average iteration time.

/* double CAnalityc::Travers ( const int position, const int typc,
const int ser, IntVect x, DblMtx p, DblMtx mu, double iter_rate) {
    double y;
    if (position == ser)
    {
        double PrConfig = 1.0, RateConfig =0.0, iter_time = 0.0;
        for (int i=0; i<ser; i++)
        {
            PrConfig = PrConfig * p[i][x[i]];
            RateConfig += mu[i][x[i]];
        }
            iter_time += PrConfig / RateConfig;
            if (iter_time > 0)
                iter_rate =  iter_rate + 1/ iter_time;
    }

    if (position < ser)
    {
        for (int i=0; i < typc; i++)
        {
            x[position] = i;
            y=iter_rate;
        iter_rate = Travers (position+1, typc, ser, x, p, mu, iter_rate);
        }
    }
    return iter_rate;
} */

void CAnalityc::Calculate(const CInput& OrigInput,std::ostream&
cLog) {
    CInput Input(OrigInput);


    int i =0, j=0,k=0,l=0;

    int n0 = Input.n0;

    int n = n0+1;

    time_t st, et;


    IntVect npr(Input.Ser,1);

    time(&st);
```

```
for(i=0; i < Input.Ser ; i++)
{

    for(j=1; j < Input.Typc; j++)
    {
        int a=0;
        for (k=0; k<j; k++)
        {

            if( (Input.Pr[i][k]==Input.Pr[i][j]) && (Input.Pr[i][k]>0) )
                a++;

        }
        if (0 == a)
            npr[i]++;
    }


}


DblVect tot(Input.Size,0.0);
DblMtx  p(Input.Ser,DblVect(Input.Typc,0.0))    ;

for (i=0; i < Input.Ser; i++)
{
    tot[i]=1.0;

    p[i][Input.Type[i]] = 1.0;
}

double s2=0, s3=0, s1=0 ,s1accu=0;

while(tot[n0] < 1.0)
{

    if(n > Input.Size)
        ExitError("huge queue");

    double s1=0, iter_time=0;
    DblVect rate(Input.Ser, 0.0);
```

```
//  Calculating service rate

        for (i=0; i<Input.Ser; i++)
        {
            iter_time=0;
            for (j=0; j<Input.Typc; j++)
            {
                iter_time += p[i][j]/Input.Mu[i][j];
            }
            rate[i] = 1/iter_time;
        }
        double rate1 = rate[0];
        double rate2 = rate[1];

        for (i=0; i<Input.Ser; i++)
            s1 += rate[i];
//  End of options,
// */

/* //   When calculating exact iteration time with Travers:

        IntVect x (Input.Ser, -1);
        s1 = Travers (0, Input.Typc, Input.Ser, x, p, Input.Mu, 0.0);
        cout << s1;
// */

        if (s1 <= 0) ExitError("s1 is not positive, very strange...");

        s2 += (1-tot[n0])/s1;
        s1accu += 1/s1;

// abandonment

        DblVect pab(Input.Typc, 0.0);
        for (int j = 0; j < Input.Typc; j++)
        {
            pab[j] = Input.Alpha[j]/(Input.Alpha[j] + s1);
        }
        for (int k = Input.Ser; (k < n) && (k < Input.Size); k++)
        {
            if ((1==tot[k]) || (k==n0))
```

```
                continue;
            tot[k] += (1-tot[k])*pab[Input.Type[k]];
        }

//   Arrivals
        DblVect ar(Input.Typc, 0.0);
        int n1;
        for (j=0; j<Input.Typc; j++)
        {
            double la1 = Input.La[0];
            ar[j] = Input.La[j]/s1;
            double ar1 = ar[0];
            double ar2 = ar[1];
            n1= n + ceil(ar[j]);
            if (n1>n)
            {
                int jj;
                for (jj=n; jj<n1; jj++)
                {
                    Input.Type[jj] = j;
                    tot[jj] = 0;
                }
            }
            tot[n1-1] = ceil(ar[j]) - ar[j];
            double totn1_1=tot[n1-1];
            n=n1;
        }


//      Probability for each server to complete service at the end of iteration
        DblVect es(Input.Ser, 0.0);
        DblMtx  p1(Input.Ser, DblVect(Input.Typc,0.0));
        for (i=0; i < Input.Ser; i++)
        {
            int a1=0;
            for (j=0; j < Input.Typc; j++)
            {
                a1 += Input.Mu[i][j] * p[i][j];
            }
            for (j=0; j < Input.Typc; j++)
            {
```

```
            p1[i][j] = p[i][j] * (1-Input.Mu[i][j]/(s1-rate[i] + Input.Mu[i][j]));
            if (p1[i][j]<0)
                    cout << endl << "p1" <<i << j << " "<<p1[i][j] << endl;
            if (p1[i][j]>1)
                    cout << endl << "p1" <<i << j << " "<<p1[i][j] << endl;

            }
            for (j=0; j< Input.Typc; j++)
            {
                es[i] += p[i][j] - p1[i][j];
                p[i][j] =p1[i][j];
            }
            if ((es[i] < 0) || (es[i] >1))
                    cout << endl << "es" << i << es[i] ;
        }


//      Assigning jobs to the available server

        for (i=0; i <Input.Ser; i++)
        {
            bool flag = 0;

            for (l=1; l<=npr[i]; l++)
            {
                if (1==flag)
                    break;


                for (k = Input.Ser; k<n; k++)
                {

                    if ((Input.Pr[i][Input.Type[k]] != 1)||(1==tot[k]))
                    {
                        continue;
                    }
                    double q = 1 - tot[k];
                    double r = p[i][Input.Type[k]];
                    double s= es[i];
                    if (q < s)
                    {
                        es[i] = s - q;
                        tot [k] = 1;
                        continue;
                    }
                    tot[k] +=s;

                    double totk = tot[k];

                    p[i][Input.Type[k]] = r + s;
                    es[i] = 0;
                    flag=1;
                    break;
                }
            }
        }
    } //while

    time(&et);
    avg_time = s2 * 60;
    elapsed_time = difftime( et, st ) * 60;


}// calculate
```

Appendix B

Simulation - C++ Program

In this appendix we present the C++ program for the simulation.

```
////////////////////////////////////////////////////////////////////////////
//class CSimulation
////////////////////////////////////////////////////////////////////////////

void CSimulation::Calculate(const CInput& OrigInput,std::ostream&
cLog) {
    const int runs = 2000;
    const double runs2 = (double)runs* (double)runs;
    time_t st, et;
    double  s3=0.0, s4=0.0, s5=0.0, p_abandon=0.0;
    cout << endl << "simulation" << endl;

    srand( (unsigned) time( NULL ) );

    time(&st);
    for (int kk=0; kk < runs; kk++)
    {
        CInput Input(OrigInput);
        int i =0, j=0, k=0, l=0;
        int n0 = Input.n0;
        int n = n0+1;
        double s1=0, s2=0;

// setting npr vector
        IntVect npr(Input.Ser,1);
        for(i=0; i < Input.Ser ; i++)
        {
            for(j=1; j < Input.Typc; j++)
            {
                int a=0;
                for (k=0; k<j; k++)
                {
                    if( (Input.Pr[i][k]==Input.Pr[i][j]) && (Input.Pr[i][k]>0) )
                        a++;
                }
                if (0 == a)
                    npr[i]++;
            }
        }

// setting P matrix and tot vector for the served customers
        DblVect tot(Input.Size,0.0);
        DblMtx  p(Input.Ser,DblVect(Input.Typc,0.0))    ;
        for (i=0; i < Input.Ser; i++)
        {
            tot[i]=1.0;

            p[i][Input.Type[i]] = 1.0;
        }
        bool end=0;
        while(0==end)
        {
            double totn0 = tot[n0];
            if(n > Input.Size)
                ExitError("huge queue");

            s1=0;
            for (i=0; i < Input.Ser; i++)
                for (j=0; j < Input.Typc; j++)
                    s1 += Input.Mu[i][j] * p[i][j];
    // t1 (iteration_time ~ Exp (s1))
            double t1 = 0;
            double x_rand = 0.0;
            while (0.0==x_rand)
                x_rand =  (double) rand() / (double) RAND_MAX;
            t1 = -log (x_rand)/s1;
            s2 += t1;

    // arrivals
            bool flag = 0;
            double t=0;
            for (j=0; j < Input.Typc; j++)
            {
                if (0 == flag)
                    t = t1;
                if (flag)
                {
                    flag = 0;
                }
                x_rand=0;
                while (0.0 == x_rand)
                    x_rand =  (double) rand() / (double) RAND_MAX;
                double x = -log (x_rand) / Input.La[j];
                if ( x > t )
                    continue;
                Input.Type[n] = j;
                tot[n] = 0;
                n++;
                t -=x;
                flag = 1;
                j--;
            }

    // abandons
            for (k=Input.Ser; k < n; k++)
            {
                if (k==n0)
                    continue;
                if (tot[k] < 1)
                {
                    x_rand =  (double) rand() / (double) RAND_MAX;
                    p_abandon = 1 - 1/exp(Input.Alpha[Input.Type[k]]*t1);
                    if (x_rand < p_abandon)
                        tot[k] = 1;
                }

            }


    // Service Completion
            DblVect es(Input.Ser, 0.0);
            int serv = -1;
```

```
for (i=0; i < Input.Ser; i++)
{
    es[i] = 0;
    for (j=0; j < Input.Typc; j++)
        es[i] += p[i][j] * Input.Mu[i][j] / s1;
}
double es0 = es[0];
double es1 = es[1];
x_rand =  (double) rand() / (double) RAND_MAX;
for (i=0; i < Input.Ser; i++)
{
    if (x_rand <= es[i])
    {
        serv = i;
        break;
    }

    x_rand -= es[i];
    serv = Input.Ser - 1;
}

bool  next_iteration = 0;
for (l=1; l <= npr[serv]; l++)
{
    if (next_iteration)
        break;

    if (end)
        break;

    for (k=Input.Ser; k < n; k++)
    {
        if ((Input.Pr[serv][Input.Type[k]] != l) || (1==tot[k]))
            continue;

        if (k == n0)
        {
            end = 1;
            break;
        }
        for (j=0; j < Input.Typc; j++)
            p[serv][j] = 0;
```

```
                    p[serv][Input.Type[k]] = 1;
                    tot[k] = 1;
                    next_iteration = 1;
                    break;
                }
            }
        } // while

        s3 += s2;
        s4 += s2*s2;
        s5 += n;
    } //runs

    time(&et);

// Summarizing all iterations:
    avg_time = s3/runs * 60;
    elapsed_time = difftime( et, st )*60;

} //CSimulation
```

Appendix C

Numerical Results

Table C.1: Results for a balanced system with two servers $\mu_1 = 30$, $\mu_2 = 15$, $\lambda_1 = 27$

$[h]L_1^0$	L_2^0	*Expected waiting time by algorithm (run time (sec))*	*Expected waiting time by simulation (run time (sec))*	$\frac{\textit{algorithm waiting time}}{\textit{simulation waiting time}}$
0	0	1.56 (*0.1*)	1.88 (*33.0*)	0.83
0	1	4.55 (*0.1*)	4.70 (*31.0*)	0.97
0	2	7.96 (*0.1*)	7.86 (*32.0*)	1.01
0	3	11.70 (*0.1*)	11.68 (*31.0*)	1.00
0	4	15.32 (*0.1*)	15.25 (*32.0*)	1.00
0	5	18.93 (*0.1*)	19.27 (*36.0*)	0.98
0	6	22.70 (*0.1*)	22.17 (*35.0*)	1.02
0	7	26.30 (*0.1*)	26.43 (*35.0*)	1.00
0	8	29.96 (*0.1*)	29.91 (*36.0*)	1.00
0	9	33.69 (*0.1*)	33.40 (*36.0*)	1.01
0	10	37.29 (*0.1*)	37.40 (*37.0*)	1.00
1	0	3.40 (*0.1*)	3.58 (*35.0*)	0.95
1	1	6.32 (*0.1*)	6.56 (*33.0*)	0.96
1	2	9.81 (*0.1*)	9.61 (*32.0*)	1.02
1	3	13.51 (*0.1*)	13.29 (*34.0*)	1.02
1	4	17.11 (*0.1*)	16.87 (*33.0*)	1.01
1	5	20.82 (*0.1*)	20.82 (*35.0*)	1.00

$[h]L_1^0$	L_2^0	*Expected waiting time* *by algorithm* (*run time* (*sec*))	*Expected waiting time* *by simulation* (*run time* (*sec*))	$\frac{\textit{algorithm waiting time}}{\textit{simulation waiting time}}$
1	6	24.49 (*0.1*)	24.19 (*36.0*)	1.01
1	7	28.09 (*0.1*)	27.87 (*35.0*)	1.01
1	8	31.85 (*0.1*)	31.46 (*35.0*)	1.01
1	9	35.48 (*0.1*)	35.16 (*37.0*)	1.01
1	10	39.08 (*0.1*)	38.82 (*38.0*)	1.01
2	0	5.23 (*0.1*)	5.38 (*33.0*)	0.97
2	1	8.09 (*0.1*)	8.19 (*33.0*)	0.99
2	2	11.69 (*0.1*)	11.51 (*34.0*)	1.02
2	3	15.30 (*0.1*)	15.35 (*33.0*)	1.00
2	4	18.91 (*0.1*)	18.65 (*35.0*)	1.01
2	5	22.68 (*0.1*)	22.76 (*35.0*)	1.00
2	6	26.28 (*0.1*)	25.73 (*36.0*)	1.02
2	7	29.94 (*0.1*)	29.79 (*37.0*)	1.00
2	8	33.67 (*0.1*)	33.01 (*37.0*)	1.02
2	9	37.27 (*0.1*)	37.28 (*38.0*)	1.00
2	10	40.97 (*0.1*)	40.00 (*39.0*)	1.02
3	0	7.04 (*0.1*)	7.27 (*34.0*)	0.97
3	1	9.91 (*0.1*)	9.97 (*33.0*)	0.99
3	2	13.51 (*0.1*)	13.62 (*36.0*)	0.99
3	3	17.08 (*0.1*)	16.87 (*35.0*)	1.01
3	4	20.79 (*0.1*)	21.05 (*33.0*)	0.99
3	5	24.46 (*0.1*)	23.94 (*33.0*)	1.02
3	6	28.06 (*0.1*)	27.92 (*34.0*)	1.00
3	7	31.82 (*0.1*)	31.91 (*34.0*)	1.00
3	8	35.45 (*0.1*)	34.91 (*36.0*)	1.02
3	9	39.05 (*0.1*)	38.60 (*36.0*)	1.01
3	10	42.84 (*0.1*)	41.83 (*37.0*)	1.02
4	0	8.84 (*0.1*)	8.85 (*34.0*)	1.00
4	1	11.77 (*0.1*)	11.55 (*34.0*)	1.02
4	2	15.28 (*0.1*)	15.23 (*35.0*)	1.00
4	3	18.88 (*0.1*)	18.98 (*33.0*)	0.99
4	4	22.64 (*0.1*)	22.02 (*38.0*)	1.03
4	5	26.24 (*0.1*)	26.01 (*35.0*)	1.01
4	6	29.90 (*0.1*)	29.96 (*36.0*)	1.00
4	7	33.63 (*0.1*)	33.36 (*36.0*)	1.01
4	8	37.23 (*0.1*)	37.30 (*37.0*)	1.00
4	9	40.93 (*0.1*)	41.23 (*38.0*)	0.99
4	10	44.62 (*0.1*)	43.96 (*39.0*)	1.01

[h]L_1^0	L_2^0	*Expected waiting time* *by algorithm* (*run time* (*sec*))	*Expected waiting time* *by simulation* (*run time* (*sec*))	$\frac{algorithm\ waiting\ time}{simulation\ waiting\ time}$
5	0	10.63 (*0.1*)	10.83 (*33.0*)	0.98
5	1	13.62 (*0.1*)	13.59 (*33.0*)	1.00
5	2	17.05 (*0.1*)	17.23 (*35.0*)	0.99
5	3	20.75 (*0.1*)	20.33 (*35.0*)	1.02
5	4	24.42 (*0.1*)	24.36 (*35.0*)	1.00
5	5	28.01 (*0.1*)	28.30 (*35.0*)	0.99
5	6	31.77 (*0.1*)	31.73 (*37.0*)	1.00
5	7	35.40 (*0.1*)	34.93 (*36.0*)	1.01
5	8	39.01 (*0.1*)	39.94 (*38.0*)	0.98
5	9	42.79 (*0.1*)	42.34 (*38.0*)	1.01
5	10	46.39 (*0.1*)	46.19 (*40.0*)	1.00
6	0	12.48 (*0.1*)	12.85 (*34.0*)	0.97
6	1	15.44 (*0.1*)	15.37 (*35.0*)	1.00
6	2	18.87 (*0.1*)	19.15 (*34.0*)	0.99
6	3	22.62 (*0.1*)	22.22 (*34.0*)	1.02
6	4	26.21 (*0.1*)	26.74 (*35.0*)	0.98
6	5	29.88 (*0.1*)	29.71 (*36.0*)	1.01
6	6	33.60 (*0.1*)	33.24 (*37.0*)	1.01
6	7	37.20 (*0.1*)	37.32 (*38.0*)	1.00
6	8	40.90 (*0.1*)	41.33 (*38.0*)	0.99
6	9	44.59 (*0.1*)	44.62 (*41.0*)	1.00
6	10	48.19 (*0.1*)	47.60 (*43.0*)	1.01
7	0	14.32 (*0.1*)	14.49 (*36.0*)	0.99
7	1	17.21 (*0.1*)	17.22 (*34.0*)	1.00
7	2	20.75 (*0.1*)	20.83 (*33.0*)	1.00
7	3	24.41 (*0.1*)	24.54 (*33.0*)	0.99
7	4	28.00 (*0.1*)	27.55 (*35.0*)	1.02
7	5	31.76 (*0.1*)	31.36 (*34.0*)	1.01
7	6	35.39 (*0.1*)	35.44 (*36.0*)	1.00
7	7	39.00 (*0.1*)	38.40 (*39.0*)	1.02
7	8	42.78 (*0.1*)	42.83 (*40.0*)	1.00
7	9	46.38 (*0.1*)	46.52 (*39.0*)	1.00
7	10	50.03 (*0.1*)	49.58 (*40.0*)	1.01
8	0	16.14 (*0.1*)	16.56 (*39.0*)	0.97
8	1	18.98 (*0.1*)	18.87 (*34.0*)	1.01
8	2	22.62 (*0.1*)	22.52 (*36.0*)	1.00
8	3	26.19 (*0.1*)	25.93 (*35.0*)	1.01
8	4	29.85 (*0.1*)	29.86 (*36.0*)	1.00
8	5	33.57 (*0.1*)	33.35 (*37.0*)	1.01

$[h]L_1^0$	L_2^0	*Expected waiting time* *by algorithm* (*run time* (*sec*))	*Expected waiting time* *by simulation* (*run time* (*sec*))	$\frac{algorithm\ waiting\ time}{simulation\ waiting\ time}$
8	6	37.17 (*0.1*)	36.68 (*38.0*)	1.01
8	7	40.88 (*0.1*)	40.25 (*37.0*)	1.02
8	8	44.56 (*0.1*)	44.32 (*39.0*)	1.01
8	9	48.16 (*0.1*)	47.86 (*42.0*)	1.01
8	10	51.90 (*0.1*)	52.05 (*42.0*)	1.00
9	0	17.94 (*0.1*)	18.13 (*34.0*)	0.99
9	1	20.84 (*0.1*)	20.61 (*39.0*)	1.01
9	2	24.40 (*0.1*)	24.39 (*36.0*)	1.00
9	3	27.97 (*0.1*)	27.47 (*35.0*)	1.02
9	4	31.73 (*0.1*)	31.97 (*37.0*)	0.99
9	5	35.35 (*0.1*)	35.54 (*39.0*)	0.99
9	6	38.97 (*0.1*)	38.66 (*38.0*)	1.01
9	7	42.74 (*0.1*)	42.30 (*39.0*)	1.01
9	8	46.34 (*0.1*)	46.22 (*39.0*)	1.00
9	9	49.99 (*0.1*)	49.79 (*42.0*)	1.00
9	10	53.73 (*0.1*)	53.81 (*45.0*)	1.00
10	0	19.74 (*0.1*)	20.26 (*36.0*)	0.97
10	1	22.70 (*0.1*)	23.03 (*38.0*)	0.99
10	2	26.17 (*0.1*)	26.63 (*37.0*)	0.98
10	3	29.81 (*0.1*)	29.48 (*37.0*)	1.01
10	4	33.53 (*0.1*)	33.37 (*39.0*)	1.00
10	5	37.13 (*0.1*)	36.37 (*38.0*)	1.02
10	6	40.84 (*0.1*)	40.66 (*36.0*)	1.00
10	7	44.52 (*0.1*)	44.29 (*38.0*)	1.01
10	8	48.12 (*0.1*)	47.81 (*38.0*)	1.01
10	9	51.86 (*0.1*)	51.62 (*40.0*)	1.00
10	10	55.50 (*0.1*)	55.54 (*40.0*)	1.00

Table C.2: Results for a balanced system with 50 servers $\mu_1 = 30$, $\mu_2 = 15$, $\lambda_1 = 675$

[h]L_1^0	L_2^0	*Expected waiting time*	*Expected waiting time*	$\frac{\textit{algorithm waiting time}}{\textit{simulation waiting time}}$
		by algorithm	*by simulation*	
		(*run time* (*sec*))	(*run time* (*sec*))	
0	0	0.06 (*0.1*)	0.07 (*360.0*)	0.82
0	25	2.36 (*0.1*)	2.38 (*405.0*)	0.99
0	50	5.51 (*0.1*)	5.49 (*492.0*)	1.00
0	75	9.04 (*0.1*)	8.98 (*611.0*)	1.01
0	100	12.66 (*0.1*)	12.66 (*734.0*)	1.00
0	125	16.29 (*0.1*)	16.31 (*924.0*)	1.00
25	0	1.87 (*0.1*)	1.89 (*395.0*)	0.99
25	25	4.18 (*0.1*)	4.20 (*460.0*)	0.99
25	50	7.32 (*0.1*)	7.34 (*572.0*)	1.00
25	75	10.86 (*0.1*)	10.83 (*689.0*)	1.00
25	100	14.48 (*0.1*)	14.46 (*878.0*)	1.00
25	125	18.11 (*0.1*)	18.06 (*1056.0*)	1.00
50	0	3.69 (*0.1*)	3.69 (*428.0*)	1.00
50	25	5.99 (*0.1*)	5.99 (*507.0*)	1.00
50	50	9.15 (*0.1*)	9.13 (*640.0*)	1.00
50	75	12.68 (*0.1*)	12.66 (*759.0*)	1.00
50	100	16.29 (*0.1*)	16.20 (*929.0*)	1.01
50	125	19.93 (*0.1*)	19.90 (*1133.0*)	1.00
75	0	5.51 (*0.1*)	5.55 (*490.0*)	0.99
75	25	7.81 (*0.1*)	7.81 (*582.0*)	1.00
75	50	10.96 (*0.1*)	10.94 (*709.0*)	1.00
75	75	14.49 (*0.1*)	14.43 (*867.0*)	1.00
75	100	18.11 (*0.1*)	18.08 (*1060.0*)	1.00
75	125	21.75 (*0.1*)	21.80 (*1282.0*)	1.00
100	0	7.33 (*0.1*)	7.36 (*564.0*)	1.00
100	25	9.63 (*0.1*)	9.62 (*669.0*)	1.00
100	50	12.78 (*0.1*)	12.74 (*812.0*)	1.00
100	75	16.31 (*0.1*)	16.30 (*992.0*)	1.00
100	100	19.93 (*0.1*)	19.87 (*1198.0*)	1.00
100	125	23.56 (*0.1*)	23.55 (*1438.0*)	1.00
125	0	9.15 (*0.1*)	9.14 (*647.0*)	1.00
125	25	11.45 (*0.1*)	11.46 (*768.0*)	1.00
125	50	14.60 (*0.1*)	14.58 (*928.0*)	1.00
125	75	18.13 (*0.1*)	18.12 (*1163.0*)	1.00
125	100	21.75 (*0.1*)	21.70 (*1425.0*)	1.00
125	125	25.38 (*0.1*)	25.37 (*1725.0*)	1.00

Table C.3: Results for a system with 2 servers $\mu_1 = 30$, $\mu_2 = 15$, $\lambda_1 = 54$

L_1^0	L_2^0	*Expected waiting time by algorithm (run time (sec))*	*Expected waiting time by simulation (run time (sec))*	$\frac{\textit{algorithm waiting time}}{\textit{simulation waiting time}}$
0	0	9.93 (*0.1*)	10.30 (*89.0*)	0.96
0	1	31.79 (*0.1*)	26.57 (*117.0*)	1.20
0	2	53.77 (*0.1*)	44.49 (*146.0*)	1.21
0	3	75.76 (*0.1*)	66.24 (*197.0*)	1.14
0	4	97.76 (*0.1*)	84.18 (*236.0*)	1.16
0	5	119.74 (*0.1*)	102.42 (*281.0*)	1.17
0	6	141.74 (*0.1*)	127.61 (*360.0*)	1.11
0	7	163.72 (*0.1*)	140.72 (*379.0*)	1.16
0	8	185.72 (*0.1*)	166.55 (*467.0*)	1.12
0	9	207.71 (*0.1*)	183.91 (*531.0*)	1.13
0	10	229.69 (*0.1*)	205.95 (*705.0*)	1.12
1	0	19.93 (*0.1*)	19.60 (*105.0*)	1.02
1	1	41.79 (*0.1*)	36.75 (*148.0*)	1.14
1	2	63.77 (*0.1*)	53.47 (*182.0*)	1.19
1	3	85.76 (*0.1*)	77.18 (*222.0*)	1.11
1	4	107.76 (*0.1*)	100.86 (*313.0*)	1.07
1	5	129.74 (*0.1*)	117.32 (*370.0*)	1.11
1	6	151.74 (*0.1*)	136.09 (*397.0*)	1.11
1	7	173.72 (*0.1*)	152.90 (*464.0*)	1.14
1	8	195.72 (*0.1*)	170.86 (*542.0*)	1.15
1	9	217.71 (*0.1*)	192.16 (*592.0*)	1.13
1	10	239.69 (*0.1*)	213.21 (*673.0*)	1.12
2	0	29.93 (*0.1*)	30.23 (*119.0*)	0.99
2	1	51.79 (*0.1*)	46.85 (*161.0*)	1.11
2	2	73.77 (*0.1*)	66.97 (*207.0*)	1.10
2	3	95.76 (*0.1*)	85.26 (*237.0*)	1.12
2	4	117.76 (*0.1*)	106.56 (*300.0*)	1.11
2	5	139.74 (*0.1*)	125.84 (*364.0*)	1.11
2	6	161.74 (*0.1*)	142.14 (*394.0*)	1.14
2	7	183.72 (*0.1*)	159.38 (*438.0*)	1.15
2	8	205.72 (*0.1*)	189.09 (*590.0*)	1.09
2	9	227.71 (*0.1*)	205.40 (*643.0*)	1.11
2	10	249.69 (*0.1*)	226.32 (*783.0*)	1.10

L_1^0	L_2^0	*Expected waiting time*	*Expected waiting time*	$\frac{algorithm\ waiting\ time}{simulation\ waiting\ time}$
		by algorithm	*by simulation*	
		(*run time* (*sec*))	(*run time* (*sec*))	
3	0	39.93 (*0.1*)	39.64 (*154.0*)	1.01
3	1	61.79 (*0.1*)	57.59 (*182.0*)	1.07
3	2	83.77 (*0.1*)	75.84 (*221.0*)	1.10
3	3	105.76 (*0.1*)	94.50 (*271.0*)	1.12
3	4	127.76 (*0.1*)	114.67 (*337.0*)	1.11
3	5	149.74 (*0.1*)	137.73 (*410.0*)	1.09
3	6	171.74 (*0.1*)	154.29 (*446.0*)	1.11
3	7	193.72 (*0.1*)	178.90 (*545.0*)	1.08
3	8	215.72 (*0.1*)	196.98 (*597.0*)	1.10
3	9	237.71 (*0.1*)	217.16 (*676.0*)	1.09
3	10	259.69 (*0.1*)	233.50 (*772.0*)	1.11
4	0	49.93 (*0.1*)	51.99 (*171.0*)	0.96
4	1	71.79 (*0.1*)	64.21 (*190.0*)	1.12
4	2	93.77 (*0.1*)	84.84 (*246.0*)	1.11
4	3	115.76 (*0.1*)	104.57 (*323.0*)	1.11
4	4	137.76 (*0.1*)	125.69 (*368.0*)	1.10
4	5	159.74 (*0.1*)	148.07 (*429.0*)	1.08
4	6	181.74 (*0.1*)	165.32 (*482.0*)	1.10
4	7	203.72 (*0.1*)	185.87 (*560.0*)	1.10
4	8	225.72 (*0.1*)	201.59 (*627.0*)	1.12
4	9	247.71 (*0.1*)	225.14 (*777.0*)	1.10
4	10	269.69 (*0.1*)	240.76 (*804.0*)	1.12
5	0	59.93 (*0.1*)	59.21 (*225.0*)	1.01
5	1	81.79 (*0.1*)	76.12 (*267.0*)	1.07
5	2	103.77 (*0.1*)	92.57 (*264.0*)	1.12
5	3	125.76 (*0.1*)	114.73 (*314.0*)	1.10
5	4	147.76 (*0.1*)	138.31 (*398.0*)	1.07
5	5	169.74 (*0.1*)	152.65 (*434.0*)	1.11
5	6	191.74 (*0.1*)	176.29 (*503.0*)	1.09
5	7	213.72 (*0.1*)	192.12 (*558.0*)	1.11
5	8	235.72 (*0.1*)	211.77 (*638.0*)	1.11
5	9	257.71 (*0.1*)	231.15 (*719.0*)	1.11
5	10	279.69 (*0.1*)	258.56 (*853.0*)	1.08

Table C.4: Results for a system with 10 servers $\mu_1 = 30$, $\mu_2 = 15$, $\lambda_1 = 270$

L_1^0	L_2^0	*Expected waiting time* *by algorithm* (*run time* (*sec*))	*Expected waiting time* *by simulation* (*run time* (*sec*))	$\frac{\textit{algorithm waiting time}}{\textit{simulation waiting time}}$
0	0	1.28 (*0.2*)	2.25 (*40.0*)	0.57
0	5	20.13 (*0.2*)	16.80 (*103.0*)	1.20
0	10	40.13 (*0.2*)	37.87 (*261.0*)	1.06
0	15	60.13 (*0.2*)	57.09 (*426.0*)	1.05
0	20	80.13 (*2.0*)	76.74 (*666.0*)	1.04
0	25	100.13 (*3.0*)	96.71 (*922.0*)	1.04
5	0	11.28 (*0.3*)	12.43 (*86.0*)	0.91
5	5	30.13 (*0.3*)	28.71 (*185.0*)	1.05
5	10	50.13 (*0.3*)	47.04 (*333.0*)	1.07
5	15	70.13 (*1.0*)	66.27 (*523.0*)	1.06
5	20	90.13 (*2.0*)	85.89 (*785.0*)	1.05
5	25	110.13 (*4.0*)	107.16 (*1101.0*)	1.03
10	0	21.28 (*1.0*)	22.22 (*137.0*)	0.96
10	5	40.13 (*1.0*)	37.77 (*247.0*)	1.06
10	10	60.13 (*1.0*)	56.54 (*418.0*)	1.06
10	15	80.13 (*2.0*)	75.42 (*640.0*)	1.06
10	20	100.13 (*3.0*)	96.72 (*960.0*)	1.04
10	25	120.13 (*4.0*)	114.78 (*1226.0*)	1.05
15	0	31.28 (*0.5*)	32.49 (*208.0*)	0.96
15	5	50.13 (*1.0*)	47.86 (*339.0*)	1.05
15	10	70.13 (*2.0*)	66.59 (*530.0*)	1.05
15	15	90.13 (*2.0*)	85.55 (*792.0*)	1.05
15	20	110.13 (*4.0*)	106.92 (*1123.0*)	1.03
15	25	130.13 (*6.0*)	129.33 (*1544.0*)	1.01
20	0	41.28 (*1.0*)	41.86 (*283.0*)	0.99
20	5	60.13 (*2.0*)	56.50 (*416.0*)	1.06
20	10	80.13 (*2.0*)	76.93 (*666.0*)	1.04
20	15	100.13 (*3.0*)	96.55 (*963.0*)	1.04
20	20	120.13 (*5.0*)	117.26 (*1309.0*)	1.02
20	25	140.13 (*7.0*)	139.05 (*1722.0*)	1.01
25	0	51.28 (*1.0*)	52.06 (*371.0*)	0.99
25	5	70.13 (*1.0*)	67.65 (*561.0*)	1.04
25	10	90.13 (*3.0*)	88.76 (*838.0*)	1.02
25	15	110.13 (*4.0*)	109.01 (*1154.0*)	1.01
25	20	130.13 (*6.0*)	129.58 (*1573.0*)	1.00
25	25	150.13 (*7.0*)	148.07 (*1914.0*)	1.01

Table C.5: Results for a system with 20 servers $\mu_1 = 30$, $\mu_2 = 15$, $\lambda_1 = 540$

L_1^0	L_2^0	*Expected waiting time*	*Expected waiting time*	$\frac{\textit{algorithm waiting time}}{\textit{simulation waiting time}}$
		by algorithm	*by simulation*	
		(*run time* (*sec*))	(*run time* (*sec*))	
10	0	10.59 (*0.2*)	10.74 (*193.0*)	0.99
10	10	29.07 (*0.2*)	26.48 (*454.0*)	1.10
10	20	49.07 (*0.2*)	46.05 (*1052.0*)	1.07
10	30	69.07 (*0.2*)	66.07 (*2059.0*)	1.05
10	40	89.07 (*19.0*)	87.84 (*2757.0*)	1.01
10	50	109.07 (*28.0*)	105.56 (*4450.0*)	1.03
20	0	20.59 (*0.3*)	20.46 (*447.0*)	1.01
20	10	39.07 (*0.3*)	36.34 (*945.0*)	1.08
20	20	59.07 (*0.3*)	57.58 (*1903.0*)	1.03
20	30	79.07 (*21.0*)	76.97 (*2483.0*)	1.03
20	40	99.07 (*25.0*)	96.14 (*3502.0*)	1.03
20	50	119.07 (*35.0*)	115.12 (*4390.0*)	1.03
20	60	139.07 (*44.0*)	136.65 (*5859.0*)	1.02
20	70	159.07 (*57.0*)	156.26 (*7744.0*)	1.02
20	80	179.07 (*77.0*)	175.15 (*9477.0*)	1.02
20	90	199.07 (*94.0*)	198.00 (*12003.0*)	1.01
20	100	219.07 (*110.0*)	218.35 (*14437.0*)	1.00
30	0	30.59 (*3.0*)	30.99 (*551.0*)	0.99
30	10	49.07 (*6.0*)	46.11 (*1006.0*)	1.06
30	20	69.07 (*11.0*)	65.85 (*1774.0*)	1.05
30	30	89.07 (*19.0*)	85.40 (*2587.0*)	1.04
30	40	109.07 (*27.0*)	109.02 (*4072.0*)	1.00
30	50	129.07 (*39.0*)	126.34 (*5176.0*)	1.02
30	50	129.07 (*39.0*)	126.70 (*5249.0*)	1.02
30	60	149.07 (*52.0*)	144.05 (*6593.0*)	1.03
30	70	169.07 (*66.0*)	169.16 (*8861.0*)	1.00
30	80	189.07 (*82.0*)	186.95 (*10738.0*)	1.01
30	90	209.07 (*107.0*)	205.81 (*13086.0*)	1.02
30	100	229.07 (*0.5*)	227.47 (*15740.0*)	1.01
40	10	59.07 (*9.0*)	56.33 (*1335.0*)	1.05
40	20	79.07 (*16.0*)	76.72 (*2182.0*)	1.03
40	30	99.07 (*23.0*)	96.56 (*3235.0*)	1.03
40	40	119.07 (*33.0*)	117.70 (*4521.0*)	1.01
40	50	139.07 (*45.0*)	137.18 (*5958.0*)	1.01

L_1^0	L_2^0	*Expected waiting time by algorithm (run time (sec))*	*Expected waiting time by simulation (run time (sec))*	$\frac{\textit{algorithm waiting time}}{\textit{simulation waiting time}}$
50	0	50.59 (*6.0*)	50.57 (*1107.0*)	1.00
50	10	69.07 (*12.0*)	69.48 (*1875.0*)	0.99
50	20	89.07 (*19.0*)	87.31 (*2727.0*)	1.02
50	30	109.07 (*28.0*)	107.09 (*3883.0*)	1.02
50	40	129.07 (*39.0*)	126.27 (*5156.0*)	1.02
50	50	149.07 (*52.0*)	145.29 (*6603.0*)	1.03
60	0	60.59 (*9.0*)	61.56 (*1542.0*)	0.98
60	10	79.07 (*16.0*)	78.00 (*2272.0*)	1.01

Table C.6: Results for a system with 50 servers $\mu_1 = 30$, $\mu_2 = 15$, $\lambda_1 = 1350$

L_1^0	L_2^0	*Expected waiting time by algorithm (run time (sec))*	*Expected waiting time by simulation (run time (sec))*	$\frac{\textit{algorithm waiting time}}{\textit{simulation waiting time}}$
0	0	0.23 (*0.2*)	0.39 (*290.0*)	0.58
0	25	18.43 (*0.2*)	16.62 (*1072.0*)	1.11
0	50	38.43 (*0.2*)	36.99 (*2996.0*)	1.04
0	75	58.43 (*0.2*)	57.03 (*5952.0*)	1.02
0	100	78.43 (*0.2*)	76.65 (*9864.0*)	1.02
0	125	98.43 (*0.2*)	97.47 (*15082.0*)	1.01
25	0	10.23 (*0.2*)	10.49 (*702.0*)	0.98
25	25	28.43 (*0.2*)	27.68 (*2056.0*)	1.03
25	50	48.43 (*0.2*)	46.97 (*4557.0*)	1.03
25	75	68.43 (*0.2*)	66.65 (*7912.0*)	1.03
25	100	88.43 (*0.2*)	86.94 (*12754.0*)	1.02
25	125	108.43 (*0.2*)	106.35 (*17857.0*)	1.02
50	0	20.23 (*0.2*)	20.42 (*1399.0*)	0.99
50	25	38.43 (*0.2*)	37.16 (*3135.0*)	1.03
50	50	58.43 (*0.2*)	57.50 (*6192.0*)	1.02
50	75	78.43 (*0.2*)	77.50 (*10180.0*)	1.01
50	100	98.43 (*0.2*)	95.98 (*14772.0*)	1.03
50	125	118.43 (*0.2*)	117.70 (*22564.0*)	1.01
50	150	138.43 (*0.2*)	136.85 (*37398.0*)	1.01
75	0	30.23 (*0.2*)	30.58 (*2655.0*)	0.99
75	25	48.43 (*0.2*)	47.64 (*5608.0*)	1.02
75	50	68.43 (*0.2*)	66.75 (*7976.0*)	1.03
75	75	88.43 (*0.2*)	86.34 (*12374.0*)	1.02
75	100	108.43 (*0.2*)	106.07 (*18023.0*)	1.02
75	125	128.43 (*509.0*)	125.91 (*27302.4*)	1.02
75	150	148.43 (*685.0*)	146.86 (*44503.6*)	1.01
75	175	168.43 (*881.0*)	166.42 (*55512.1*)	1.01

Table C.7: Results for a system with 2 servers $\mu_1 = 30$, $\mu_2 = 15$, $\lambda_1 = 51$

L_1^0	L_2^0	*Expected waiting time by algorithm (run time (sec))*	*Expected waiting time by simulation (run time (sec))*	$\frac{algorithm\ waiting\ time}{simulation\ waiting\ time}$
$Q1$	$Q2$	*avgalgorithmtimealg*	*avgsimulationtimesim*	$\frac{algorithm}{simulation}$
0	0	6.26 (*0.1*)	5.91 (*31.0*)	1.06
0	1	20.51 (*0.1*)	17.12 (*37.0*)	1.20
0	2	34.99 (*0.1*)	31.20 (*41.0*)	1.12
0	3	49.49 (*0.1*)	43.30 (*46.0*)	1.14
0	4	63.98 (*0.1*)	55.71 (*53.0*)	1.15
1	0	12.92 (*0.1*)	13.38 (*34.0*)	0.97
1	1	27.18 (*0.1*)	24.48 (*38.0*)	1.11
1	2	41.67 (*0.1*)	38.28 (*46.0*)	1.09
1	3	56.16 (*0.1*)	50.52 (*50.0*)	1.11
1	4	70.64 (*0.1*)	62.73 (*55.0*)	1.13
2	0	19.58 (*0.1*)	21.26 (*37.0*)	0.92
2	1	33.85 (*0.1*)	31.27 (*42.0*)	1.08
2	2	48.34 (*0.1*)	43.47 (*46.0*)	1.11
2	3	62.83 (*0.1*)	56.57 (*53.0*)	1.11
2	4	77.31 (*0.1*)	70.60 (*61.0*)	1.10
3	0	26.26 (*0.1*)	27.33 (*40.0*)	0.96
3	1	40.51 (*0.1*)	37.30 (*43.0*)	1.09
3	2	54.99 (*0.1*)	49.29 (*47.0*)	1.12
3	3	69.49 (*0.1*)	61.30 (*56.0*)	1.13
3	4	83.98 (*0.1*)	76.75 (*63.0*)	1.09
4	0	32.92 (*0.1*)	31.98 (*40.0*)	1.03
4	1	47.18 (*0.1*)	44.47 (*48.0*)	1.06
4	2	61.67 (*0.1*)	55.87 (*57.0*)	1.10
4	3	76.16 (*0.1*)	69.12 (*64.0*)	1.10
4	4	90.64 (*0.1*)	83.86 (*74.0*)	1.08
5	0	39.58 (*0.1*)	38.95 (*45.0*)	1.02
5	1	53.85 (*0.1*)	53.36 (*53.0*)	1.01
5	2	68.34 (*0.1*)	62.82 (*58.0*)	1.09
5	3	82.83 (*0.1*)	78.37 (*71.0*)	1.06
5	4	97.31 (*0.1*)	91.39 (*81.0*)	1.06

L_1^0	L_2^0	*Expected waiting time*	*Expected waiting time*	$\frac{algorithm\ waiting\ time}{simulation\ waiting\ time}$
		by algorithm	*by simulation*	
		(*run time* (*sec*))	(*run time* (*sec*))	
6	0	46.26 (*0.1*)	47.45 (*50.0*)	0.97
6	1	60.51 (*0.1*)	59.22 (*56.0*)	1.02
6	2	74.99 (*0.1*)	67.91 (*60.0*)	1.10
6	3	89.49 (*0.1*)	82.19 (*75.0*)	1.09
6	4	103.98 (*0.1*)	98.13 (*92.0*)	1.06
7	0	52.92 (*0.1*)	51.54 (*51.0*)	1.03
7	1	67.18 (*0.1*)	66.12 (*60.0*)	1.02
7	2	81.67 (*0.1*)	79.35 (*70.0*)	1.03
7	3	96.16 (*0.1*)	91.30 (*78.0*)	1.05
7	4	110.64 (*0.1*)	104.73 (*87.0*)	1.06
8	0	59.58 (*0.1*)	58.67 (*54.0*)	1.02
8	1	73.85 (*0.1*)	70.82 (*63.0*)	1.04
8	2	88.34 (*0.1*)	84.52 (*70.0*)	1.05
8	3	102.83 (*0.1*)	101.09 (*86.0*)	1.02
8	4	117.31 (*0.1*)	114.61 (*99.0*)	1.02
9	0	66.26 (*0.1*)	67.33 (*60.0*)	0.98
9	1	80.51 (*0.1*)	76.42 (*67.0*)	1.05
9	2	94.99 (*0.1*)	92.31 (*78.0*)	1.03
9	3	109.49 (*0.1*)	105.39 (*88.0*)	1.04
9	4	123.98 (*0.1*)	118.58 (*102.0*)	1.05
10	0	72.92 (*0.1*)	76.11 (*68.0*)	0.96
10	1	87.18 (*0.1*)	83.67 (*71.0*)	1.04
10	2	101.67 (*0.1*)	98.67 (*84.0*)	1.03
10	3	116.16 (*0.1*)	110.57 (*92.0*)	1.05
10	4	130.64 (*0.1*)	123.96 (*101.0*)	1.05

Table C.8: Results for a system with 2 servers $\mu_1 = 15$, $\mu_2 = 30$, $\lambda_1 = 25.5$

L_1^0	L_2^0	*Expected waiting time by algorithm (run time (sec))*	*Expected waiting time by simulation (run time (sec))*	$\frac{\textit{algorithm waiting time}}{\textit{simulation waiting time}}$
0	0	6.11 (*0.0*)	14.37 (*96.0*)	0.43
0	1	14.01 (*0.0*)	21.89 (*96.0*)	0.64
0	2	21.33 (*0.0*)	29.85 (*104.0*)	0.71
0	3	28.58 (*0.0*)	35.79 (*118.0*)	0.80
0	4	35.83 (*0.0*)	43.92 (*114.0*)	0.82
0	5	43.07 (*0.0*)	47.88 (*121.0*)	0.90
0	6	50.31 (*0.0*)	57.03 (*137.0*)	0.88
0	7	57.56 (*0.0*)	62.85 (*156.0*)	0.92
0	8	64.80 (*0.0*)	67.94 (*147.0*)	0.95
0	9	72.04 (*0.0*)	75.88 (*164.0*)	0.95
0	10	79.29 (*0.0*)	83.35 (*179.0*)	0.95
1	0	19.43 (*0.0*)	27.15 (*108.0*)	0.72
1	1	27.34 (*0.0*)	34.34 (*113.0*)	0.80
1	2	34.66 (*0.0*)	42.75 (*125.0*)	0.81
1	3	41.91 (*0.0*)	49.68 (*125.0*)	0.84
1	4	49.15 (*0.0*)	55.83 (*137.0*)	0.88
1	5	56.40 (*0.0*)	62.43 (*134.0*)	0.90
1	6	63.64 (*0.0*)	68.45 (*151.0*)	0.93
1	7	70.88 (*0.0*)	75.43 (*145.0*)	0.94
1	8	78.13 (*0.0*)	83.08 (*152.0*)	0.94
1	9	85.38 (*0.0*)	90.03 (*170.0*)	0.95
1	10	92.63 (*0.0*)	99.81 (*191.0*)	0.93
2	0	32.76 (*0.0*)	40.26 (*112.0*)	0.81
2	1	40.69 (*0.0*)	47.77 (*116.0*)	0.85
2	2	48.00 (*0.0*)	56.25 (*132.0*)	0.85
2	3	55.25 (*0.0*)	63.11 (*159.0*)	0.88
2	4	62.49 (*0.0*)	69.88 (*154.0*)	0.89
2	5	69.74 (*0.0*)	75.83 (*155.0*)	0.92
2	6	76.98 (*0.0*)	82.36 (*173.0*)	0.93
2	7	84.22 (*0.0*)	89.12 (*183.0*)	0.95
2	8	91.47 (*0.0*)	96.42 (*189.0*)	0.95
2	9	98.71 (*0.0*)	104.04 (*215.0*)	0.95
2	10	105.95 (*0.0*)	109.30 (*197.0*)	0.97

L_1^0	L_2^0	*Expected waiting time by algorithm (run time (sec))*	*Expected waiting time by simulation (run time (sec))*	$\frac{\textit{algorithm waiting time}}{\textit{simulation waiting time}}$
3	0	46.11 (*0.0*)	53.03 (*133.0*)	0.87
3	1	54.01 (*0.0*)	61.13 (*132.0*)	0.88
3	2	61.33 (*0.0*)	70.45 (*140.0*)	0.87
3	3	68.58 (*0.0*)	77.78 (*140.0*)	0.88
3	4	75.83 (*0.0*)	83.95 (*159.0*)	0.90
3	5	83.07 (*0.0*)	88.91 (*157.0*)	0.93
3	6	90.31 (*0.0*)	98.16 (*173.0*)	0.92
3	7	97.56 (*0.0*)	100.93 (*170.0*)	0.97
3	8	104.80 (*0.0*)	109.23 (*181.0*)	0.96
3	9	112.05 (*0.0*)	117.07 (*209.0*)	0.96
3	10	119.29 (*0.0*)	124.14 (*211.0*)	0.96
4	0	59.43 (*0.0*)	66.28 (*131.0*)	0.90
4	1	67.34 (*0.0*)	76.52 (*146.0*)	0.88
4	2	74.66 (*0.0*)	83.51 (*148.0*)	0.89
4	3	81.91 (*0.0*)	90.95 (*160.0*)	0.90
4	4	89.15 (*0.0*)	94.87 (*166.0*)	0.94
4	5	96.40 (*0.0*)	101.99 (*180.0*)	0.95
4	6	103.64 (*0.0*)	111.35 (*193.0*)	0.93
4	7	110.88 (*0.0*)	116.76 (*196.0*)	0.95
4	8	118.13 (*0.0*)	124.08 (*213.0*)	0.95
4	9	125.38 (*0.0*)	128.23 (*221.0*)	0.98
4	10	132.63 (*0.0*)	134.84 (*226.0*)	0.98
5	0	72.76 (*0.0*)	77.93 (*146.0*)	0.93
5	1	80.69 (*0.0*)	89.39 (*171.0*)	0.90
5	2	88.00 (*0.0*)	93.98 (*178.0*)	0.94
5	3	95.25 (*0.0*)	105.58 (*180.0*)	0.90
5	4	102.49 (*0.0*)	110.05 (*180.0*)	0.93
5	5	109.74 (*0.0*)	116.56 (*199.0*)	0.94
5	6	116.98 (*0.0*)	121.90 (*207.0*)	0.96
5	7	124.22 (*0.0*)	133.12 (*241.0*)	0.93
5	8	131.47 (*0.0*)	137.48 (*252.0*)	0.96
5	9	138.71 (*0.0*)	143.51 (*254.0*)	0.97
5	10	145.95 (*0.0*)	149.80 (*283.0*)	0.97

Table C.9: Results for high priority waiting times in a system with 2 servers $\mu_1 = 15$, $\mu_2 = 30$, $\lambda_1 = 25.5$

$[h]L_1^0$	L_2^0	*Expected waiting time by algorithm (run time (sec))*	*Expected waiting time by simulation (run time (sec))*	$\frac{\textit{algorithm waiting time}}{\textit{simulation waiting time}}$
0	0	2.00 (*0.1*)	2.01 (*66.0*)	1.00
0	5	2.00 (*0.1*)	2.02 (*65.0*)	0.99
0	10	2.00 (*0.1*)	2.01 (*65.0*)	1.00
0	15	2.00 (*0.1*)	2.01 (*65.0*)	1.00
0	20	2.00 (*0.1*)	2.01 (*71.0*)	1.00
5	0	12.00 (*0.1*)	12.05 (*67.0*)	1.00
5	5	12.00 (*0.1*)	11.95 (*69.0*)	1.00
5	10	12.00 (*0.1*)	12.01 (*68.0*)	1.00
5	15	12.00 (*0.1*)	11.96 (*69.0*)	1.00
5	20	12.00 (*0.1*)	11.98 (*68.0*)	1.00
10	0	22.00 (*0.1*)	22.03 (*69.0*)	1.00
10	5	22.00 (*0.1*)	22.04 (*70.0*)	1.00
10	10	22.00 (*0.1*)	21.98 (*70.0*)	1.00
10	15	22.00 (*0.1*)	21.92 (*70.0*)	1.00
10	20	22.00 (*0.1*)	21.98 (*71.0*)	1.00
15	0	32.00 (*0.1*)	31.89 (*72.0*)	1.00
15	5	32.00 (*0.1*)	31.97 (*72.0*)	1.00
15	10	32.00 (*0.1*)	32.14 (*75.0*)	1.00
15	15	32.00 (*0.1*)	31.99 (*73.0*)	1.00
15	20	32.00 (*0.1*)	32.09 (*72.0*)	1.00
20	0	42.00 (*0.2*)	41.91 (*72.0*)	1.00
20	5	42.00 (*0.2*)	41.97 (*73.0*)	1.00
20	10	42.00 (*0.2*)	41.68 (*79.0*)	1.01
20	15	42.00 (*0.2*)	42.16 (*78.0*)	1.00
20	20	42.00 (*0.2*)	42.17 (*75.0*)	1.00

Table C.10: Results for a system with abandonment: 50 servers $\mu_1 = 30$, $\mu_2 = 15$, $\lambda_1 = 1350$, $\alpha_1 = 12$, $\alpha_2 = 12$

L_1^0	L_2^0	*Expected waiting time*	*Expected waiting time*	$\frac{\textit{algorithm waiting time}}{\textit{simulation waiting time}}$
		by algorithm	*by simulation*	
		(*run time* (*sec*))	(*run time* (*sec*))	
0	0	0.23 (*0.1*)	0.25 (*361.0*)	0.92
0	25	7.86 (*0.1*)	6.14 (*553.0*)	1.28
0	50	11.09 (*0.1*)	9.19 (*685.0*)	1.21
0	75	13.04 (*0.1*)	11.18 (*811.0*)	1.17
0	100	14.45 (*0.1*)	12.41 (*868.0*)	1.16
0	125	15.55 (*0.1*)	13.58 (*952.0*)	1.15
0	150	16.45 (*0.1*)	14.37 (*1026.0*)	1.14
0	175	17.21 (*0.1*)	15.09 (*1101.0*)	1.14
0	200	17.88 (*0.1*)	15.95 (*1191.0*)	1.12
0	225	18.46 (*0.1*)	16.50 (*1259.0*)	1.12
0	250	18.99 (*0.1*)	17.09 (*1336.0*)	1.11
25	0	5.74 (*0.1*)	4.70 (*468.0*)	1.22
25	25	9.15 (*0.1*)	7.61 (*593.0*)	1.20
25	50	11.64 (*0.1*)	9.86 (*713.0*)	1.18
25	75	13.38 (*0.1*)	11.57 (*823.0*)	1.16
25	100	14.69 (*0.1*)	12.79 (*913.0*)	1.15
25	125	15.73 (*0.1*)	13.92 (*1009.0*)	1.13
25	150	16.60 (*0.1*)	14.70 (*1088.0*)	1.13
25	175	17.34 (*0.1*)	15.40 (*1161.0*)	1.13
25	200	17.99 (*0.1*)	16.00 (*1234.0*)	1.12
25	225	18.56 (*0.1*)	16.57 (*1307.0*)	1.12
25	250	19.07 (*0.1*)	17.16 (*1382.0*)	1.11
50	0	8.31 (*0.1*)	7.08 (*569.0*)	1.17
50	25	10.43 (*0.1*)	8.98 (*672.0*)	1.16
50	50	12.29 (*0.1*)	10.58 (*774.0*)	1.16
50	75	13.78 (*0.1*)	11.95 (*871.0*)	1.15
50	100	14.97 (*0.1*)	13.15 (*964.0*)	1.14
50	125	15.94 (*0.1*)	14.13 (*1051.0*)	1.13
50	150	16.77 (*0.1*)	14.81 (*1125.0*)	1.13
50	175	17.48 (*0.1*)	15.55 (*1203.0*)	1.12
50	200	18.11 (*0.1*)	16.14 (*1278.0*)	1.12
50	225	18.66 (*0.1*)	16.69 (*1348.0*)	1.12
50	250	19.17 (*0.1*)	17.33 (*1428.0*)	1.11

L_1^0	L_2^0	*Expected waiting time*	*Expected waiting time*	$\frac{\textit{algorithm waiting time}}{\textit{simulation waiting time}}$
		by algorithm	*by simulation*	
		(*run time* (*sec*))	(*run time* (*sec*))	
75	0	9.99 (*0.1*)	8.69 (*660.0*)	1.15
75	25	11.52 (*0.1*)	10.05 (*746.0*)	1.15
75	50	12.96 (*0.1*)	11.29 (*836.0*)	1.15
75	75	14.22 (*0.1*)	12.42 (*926.0*)	1.14
75	100	15.28 (*0.1*)	13.50 (*1028.0*)	1.13
75	125	16.18 (*0.1*)	14.21 (*1089.0*)	1.14
75	150	16.95 (*0.1*)	15.08 (*1176.0*)	1.12
75	175	17.63 (*0.1*)	15.66 (*1247.0*)	1.13
75	200	18.23 (*0.1*)	16.38 (*1334.0*)	1.11
75	225	18.78 (*0.1*)	16.75 (*1391.0*)	1.12
75	250	19.26 (*0.1*)	17.39 (*2081.0*)	1.11
100	0	11.26 (*0.1*)	9.84 (*830.0*)	1.14
100	25	12.45 (*0.1*)	10.93 (*912.0*)	1.14
100	50	13.61 (*0.1*)	12.01 (*1003.0*)	1.13
100	75	14.67 (*0.1*)	13.02 (*1080.0*)	1.13
100	100	15.60 (*0.1*)	13.81 (*1213.0*)	1.13
100	125	16.43 (*0.1*)	14.69 (*1274.0*)	1.12
100	150	17.15 (*0.1*)	15.28 (*1248.0*)	1.12
100	175	17.80 (*0.1*)	15.99 (*1333.0*)	1.11
100	200	18.37 (*0.1*)	16.39 (*1399.0*)	1.12
100	225	18.89 (*0.1*)	16.95 (*1465.0*)	1.11
100	250	19.37 (*0.1*)	17.44 (*1614.0*)	1.11
125	0	12.26 (*0.1*)	10.88 (*848.0*)	1.13
125	25	13.24 (*0.1*)	11.67 (*894.0*)	1.13
125	50	14.20 (*0.1*)	12.57 (*990.0*)	1.13
125	75	15.11 (*0.1*)	13.47 (*1089.0*)	1.12
125	100	15.94 (*0.1*)	14.17 (*1152.0*)	1.12
125	125	16.69 (*0.1*)	14.95 (*1211.0*)	1.12
125	150	17.36 (*0.1*)	15.54 (*1292.0*)	1.12
125	175	17.97 (*0.1*)	16.25 (*1381.0*)	1.11
125	200	18.52 (*0.1*)	16.67 (*1598.0*)	1.11
125	225	19.02 (*0.1*)	17.11 (*1476.0*)	1.11
125	250	19.48 (*0.1*)	17.52 (*1541.0*)	1.11

Bibliography

[1] S. Borst, A. Mandelbaum, and M.I. Reiman. Dimensioning large call centers. Submitted for publication, 2001.

[2] Z. Carmon and D. Kahenman. The experienced utility of queueing: Experience profiles and retrospective evaluatoins of simulated queues. pre-print.

[3] Z. Carmon, J.G. Shanthikumar, and T.F. Carmon. A psychological perspective on service segmentation models: The significance of accounting for consumers' perceptions of waiting and service. *Management Science*, 41(11):1806–1815, 1995.

[4] B. Cleveland and J. Mayben. *Call Center Management on Fast Forward: Succeeding in Today's Dynamic Inbound Environment*. Call Center Press, 1999.

[5] A. Cobham. Priority assignment in waiting line problems. *Operations Evaluation Group, United States Navy*, 1953.

[6] M.M. Davis. How long should a customer wait for service? *Decision Sciences*, 22:421–434, 1991.

[7] D. Duxbruy, R. Backhouse, M. Head, G. Llyod, and J. Pilkington. Call centers in bt uk customer service. *British Telecommunication Engineering*, 18:165–173, 1999.

[8] R.A. Feinberg, I. Kim, and L. Hokama. Operational determinants of caller satisfaction in the call center. *International Journal of Service Industry Management*, 11(2):131–141, 2000.

[9] O. Garnett and A. Mandelbaum. An introduction to skills-based routing and its operational compelxities. Teaching-note, Service Engineering, Technion, Israel, 2000.

[10] O. Garnett, A. Mandelbaum, and M. Reiman. Desigining a call center with impatient customers. *preprint*, 1999. Available at http://ie.technion.ac.il/serveng.

[11] L. Green. A queueing system with general-use and limited-use servers. *Columbia University, New York, New York*, 1984.

[12] S. Halfin and W. Whitt. Heavy-traffic climaitis for queues with many exponenial servers. *Operations research*, 29:567–587, 1981.

[13] M. K. Hui and D. K. Tse. What to tell consumers in waits of different lengths: an integrative model of service evaluation. *Journal of Marketing*, 60:81–90, 1996.

[14] K. Katz, B. Larson, and R. Larson. Prescription for the waiting-in-line blues: Entertain, enlighten, and engage. *Sloan Management Review*, pages 44–53, Winter 1991.

[15] G. Koole and A. Mandelbaum. Queuing models of call centers - an introductoin. 2001.

[16] R.C. Larson. Perspectives on queues: social justice and the psychology of queueing. *Operations Research*, 35(6):895–905, 1987.

[17] D.H. Maister. The psychology of waiting lines. In J.A. Czepiel et. al., editor, *The Service Encounter*. Lexington Books, 1985.

[18] A. Mandelbaum. Call centers, research bibliography with absracts. 2001.

[19] A. Mandelbaum, W.A.M Massey, M.I. Reiman, and R. Rider. Time varing multiserver queues with abandonment and retrials. In *In P. Key and D. Smith, editors,Proceedings of the 16th International Teletraffic Conference*, 1999.

[20] A. Mandelbaum, W.A.M Massey, M.I. Reiman, R. Rider, and A. Stoylar. Queue lengths and waiting times for multiserver queues with abandonment and retrials. *Working paper*, 2000.

[21] A. Mandelbaum, A. Sakov, and S. Zeltyn. Empirical analysis of a call center. *Technical Report, Technion*, August 2000.

[22] R. Nelson. *Probability, Stochastic Processes, and Queuing Theory*. Springer-Verlage, 1995.

[23] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models*. The Johns Hopkins University Press, 1981.

[24] M. Perry and A. Nillson. Performance modeling of automatic call distributors: operaor services staffing with heterogeneous positions. *Fundamental role of teletraffic in the evolution of telecommunication networks. Proceeding of the 14th intenational congress*, ITC-14. Elsevier, Amsterdam, The Netherlands:1023–1032, 1994.

[25] A.A. Puhalskii and M.I. Reiman. The multiclass gi/ph/n queue in the halfin whitt regime. *Advences in applied probability*, 32:564–595, 2000.

[26] D.R. Roque. Technical notes: A note on "Queueing models with lane selection". *Operation Research*, 28(2):419–420, 1980.

[27] B.L. Schwartz. Queueing models with lane selection: a new class of problems. *Operation Research*, 22:331–339, 1974.

[28] D.Y. Sze. A queueing model for telephone operator staffing. *Operations Research*, 32(2):229–249, 1984.

[29] S. Taylor. Waiting for service: the relationship between delays and evaluation of service. *Journal of Marketing*, 58:56–69, 1994.

[30] G. Tom, M. Burns, and Y. Zeng. Your life on hold; The effect of telephone waiting time on customer perception. *Journal of Direct Marketing*, 11(3):25–31, 1997.

[31] W. Whitt. Improving service by informing customers about anticipated delays. *Management Science*, 45 (2):192–207, 1999.

[32] W. Whitt. Partitioning customers into service groups. *Management Science*, 45(11):1579–1592, 1999.

[33] W. Whitt. Predicting queueing delays. *Management Science*, 45 (6):870–888, 1999.

[34] C.M. Woodside, D.A. Stanford, and B. Pagurek. Optimal prediction of queue lengths and delays in gi/m/m multiserver queues. *Operations Research*, 32(4):809–817, 1984.

[35] E. Zohar, A. Mandelbaum, and N. Shimkin. Adaptive behavior of impatient customers in telequeues: Theory and empirical support. *Prepring*, 2000.