The Power of Two Choices in Randomized Load Balancing

by

Michael David Mitzenmacher

B.A. (Harvard University) 1991

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION
of the
UNIVERSITY of CALIFORNIA at BERKELEY

Committee in charge:

Professor Alistair Sinclair, Chair
Professor Christos Papadimitriou
Professor David Aldous

1996

The dissertation of Michael David Mitzenmacher is approved:

Chair Date

Date

Date

University of California at Berkeley

1996

The Power of Two Choices in Randomized Load Balancing

by

Michael David Mitzenmacher

Abstract

The Power of Two Choices in Randomized Load Balancing

by

Michael David Mitzenmacher
Doctor of Philosophy in Computer Science

University of California at Berkeley

Professor Alistair Sinclair, Chair

Suppose that n balls are placed into n bins, each ball being placed into a bin chosen independently and uniformly at random. Then, with high probability, the maximum load in any bin is approximately $\frac{\log n}{\log \log n}$. Suppose instead that each ball is placed sequentially into the least full of d bins chosen independently and uniformly at random. It has recently been shown that the maximum load is then only $\frac{\log \log n}{\log d} + O(1)$ with high probability. Thus giving each ball two choices instead of just one leads to an exponential improvement in the maximum load. This result demonstrates the power of two choices, and it has several applications to load balancing in distributed systems.

In this thesis, we expand upon this result by examining related models and by developing techniques for studying similar randomized load balancing schemes. We first remove the restriction above that the balls be placed sequentially. We provide a lower bound demonstrating a tradeoff between the number of rounds of communication used and the maximum load achieved. Our lower bound shows that one cannot achieve a maximum load of $O(\log \log n)$ with only a constant number of rounds of communication. We also provide simple algorithms that match our lower bounds within a constant factor.

We then consider dynamic models, where balls enter and leave the system over time. This leads to a natural queueing problem: customers arrive as a Poisson stream at a collection of n servers. Each customer chooses d of these servers independently and uniformly at random and queues at the server currently containing the fewest customers. Customers require an exponentially distributed amount of service time before leaving the system. We call this model the *supermarket model*. We determine the behavior of the supermarket model by defining an idealized process, corresponding to a system of infinite size, which is cleaner and easier to analyze. We then relate the idealized system to the finite system, bounding the error between them. Using this technique, we also study several generalizations of the supermarket model and many other load balancing schemes. Our results demonstrate the effectiveness of having two choices in many situations.

Professor Alistair Sinclair
Dissertation Committee Chair

To all the teachers

who have helped me reach this far.

Contents

List of Figures

List of Tables

Acknowledgments

This thesis could not have come about without the assistance and encouragement of many people. The most important of these is my advisor, Alistair Sinclair. His helpful ideas and motivating discussions have impacted and improved this research in countless ways. I cannot imagine my years here without the benefit of his guidance and support.

I would also like to thank Andrei Broder, with whom I worked during a summer internship at Digital Systems Research Center. Besides assisting me in my research, Andrei inspired me to finish my thesis.

The infectious enthusiasm of the Berkeley faculty have made it an exciting place to be. Richard Karp, Umesh Vazirani, Raimund Seidel, Christos Papadimitriou, Michael Luby, David Aldous, and Manuel Blum have always been there to answer questions or to offer an encouraging word when it was needed.

The other students at Berkeley have made my research and my extracurricular life much richer. Steve Lumetta, who always listened to my craziest ideas, often helped me to see things more clearly. Micah Adler, Soumen Chakrabarti, and Lars Rasmussen deserve special recognition, both for their part in the work that we did together, and for always being there to talk about problems. Diane Hernek and Dana Randall, my unofficial big siblings, gave me useful advice often over the years. John Byers, David Blackston, Debbie Weisser, Eric Vigoda, Jeff Erickson, and all the others I have had the pleasure of working and playing with all have my deepest appreciation.

Finally, I would like to thank my family, who always believed I could.

This work was supported in part by an Office of Naval Research Fellowship and by NSF Grant CCR-9505448.

Chapter 1

Introduction

1.1 Load balancing problems

Load balancing is the act of distributing objects among a set of locations as evenly as possible. For example, suppose one has a set of tasks S to distribute among a set of processors P, such that only one task can run on a processor at any time, a task must be run entirely on a single processor, tasks cannot be preempted, and processors compute at the same rate. Given the execution time of each task, how should one distribute them to minimize the final completion time? This problem is motivated by the increasing use of parallel machines to run large programs, many of which can be broken down into smaller subprocesses to be divided among the processors.

Surprisingly, even in the case where there are only two processors, finding the exact answer to this question is an NP-complete problem, and hence, presumably, computationally intractable. In fact, this was one of the original problems shown to be NP-complete by Karp [33, p. 238]. Because in general even simple load balancing problems such as this prove computationally intractable, one is often forced to look for effective strategies that provide suitable performance in practice. For example, one may consider load balancing in an *on-line* setting, where tasks appear one at a time and must be placed upon arrival. In the on-line setting, achieving the optimal load balance for a set of tasks is generally impossible, since one does not know what the future will bring. Hence, one's goal must change to finding an algorithm that does well over all (or most) inputs. Another possibility, which can lead to more accurate estimates of the behavior of an algorithm, is to look at *average-case analysis*, if the execution times vary according to a fixed, known probability distribution. In average-case analysis, one attempts to make statements about the distribution of completion time of an algorithm based on assumptions about the distribution of the input parameters, such as the number of jobs and the execution times. Most of the work of this thesis will focus on on-line algorithms and average-case analysis.

In developing simple, effective load balancing algorithms, *randomization* often proves to be a useful tool. For example, if we have a set of tasks S and processors P, one possible way to distribute the tasks is to simply place each task on a random processor, that is, a processor chosen independently and uniformly at random. With this strategy, the *expected load* at each processor is the same, and hence intuitively, if there are enough tasks and the task sizes are not too disparate, then this strategy should load the processors almost equally.

In this thesis, we will examine the effectiveness of this and other similar randomized load balancing strategies. Our emphasis will be the tradeoff between the amount of coordination and the distribution of load. The simple randomized strategy described above requires no coordination among the tasks or the processors. We will show that with just a small amount of coordination, one can often achieve a much better distribution of load.

1.1.1 The balls and bins model

The best way to describe the sort of problems we will be studying is to begin with some examples. We adopt a simplified model historically used by mathematicians: the balls and bins model, *e.g.* [40, 47]. Suppose that n balls are placed into n bins, with each ball being placed into a bin chosen independently and uniformly at random. Let the *load* of a bin be the number of balls in that bin after all balls have been thrown. What is the *maximum load* over all the bins once the process terminates? It is well known that with high probability, the maximum load upon completion will be approximately $\frac{\log n}{\log \log n}$ [35].[1] In the processor model mentioned above, this says that if we distribute n tasks to n processors randomly, with each task requiring unit time, then all tasks will complete in time $\frac{\log n}{\log \log n}$ with high probability.

Clearly we can do better. If we did not place balls randomly, then we could easily distribute the tasks so that exactly one ball ends up in each bin. It is therefore

[1] Throughout this thesis, when we say *with high probability* we shall mean with probability at least $1 - O(1/n)$, where n is the number of balls. Also, log will always mean the natural logarithm, unless otherwise noted.

not immediately clear that this result has any practical significance; however, it actually has many applications. For example, suppose we hash n items into a hash table of n buckets, assuming that the results of the hashes yield an independent random location for each item. Then the above result shows that, with high probability, the largest bucket has approximately $\frac{\log n}{\log \log n}$ entries, and hence the search time for an item in the hash table will be $O(\frac{\log n}{\log \log n})$.[2]

The result also has more subtle implications in the processor model. Placing one ball in each bin requires a *centralized* system, where either some agent controls the destination bin of each ball, or the balls communicate in order to agree on a placement strategy. Consider instead a *distributed* system, where the tasks may not have any direct communication (or where such communication is expensive). In a distributed system, placing the balls randomly may be appropriate. There is a tradeoff between the maximum load and the communication required: a maximum load of 1 with complete coordination versus a maximum load of $\frac{\log n}{\log \log n}$ with no coordination. Given this tradeoff in the extreme cases, it makes sense to consider strategies with limited communication, or partial coordination.

1.1.2 The power of two choices

We now state a surprising result, proven in a seminal paper by Azar, Broder, Karlin, and Upfal [11]. (We shall present a proof of this result in Section 1.2 of this introduction.) Suppose that the balls are placed sequentially, so that for each ball we choose *two* bins independently and uniformly at random and place the ball into the less full bin (breaking ties arbitrarily). In this case, the maximum load drops to $\frac{\log \log n}{\log 2} + O(1)$ with high probability. If each ball instead has d choices, then the maximum load will be $\frac{\log \log n}{\log d} + O(1)$ with high probability. Having two choices hence yields a qualitatively different type of behavior from the single choice case, leading to an exponential improvement in the maximum load; having more than two choices further improves the maximum load by only a constant factor. Following Azar *et al.*, we refer to this algorithm in which each ball has d choices as GREEDY(d).

[2] The same result clearly holds even if the hashes are not completely independent, but only $O(\frac{\log n}{\log \log n})$-wise independent, since this does not affect the probability that $O(\frac{\log n}{\log \log n})$ balls are in the same bin. In this thesis we make the simplifying assumption that random choices are completely independent. In most cases, the amount of randomness required can be reduced in a straightforward manner up to a certain point; however, the development of suitable hash functions using only small amounts of randomness is an interesting subject. See [42] for a powerful family of hash functions, or [54] for an introduction to limited independence and universal families of hash functions.

Again, this result has some remarkable practical implications. For example, if one hashes n items sequentially into n buckets using two hash functions, placing an item only in the bucket with the smaller load, then by using both hash functions to search for an item, one reduces the maximum time to find an item to $O(\log\log n)$ with high probability. If we use this strategy in the processor model where all tasks take unit time, then the time until termination drops to only $O(\log\log n)$, with only a small amount of coordination.

At a more basic level, this result demonstrates the power even a small amount of additional information can have on a natural performance measure. The gain from each ball having two choices is dramatic, but more choices prove much less significant: the law of diminishing returns appears remarkably strong in this scenario!

1.1.3 Questions to address

A natural question to ask is whether this fundamental idea, *the power of two choices*, is a paradigm that applies to many situations, or is merely an artifact of the idealized models.

Question 1: What is the key feature that leads to the $\log\log n + O(1)$ bounds? Does this feature appear in other systems?

Upon reflection, other questions also arise from this result and its proof. Indeed, like most interesting results, this one leaves at least as many questions as it does answers. Here we discuss some of the most striking. A first natural question regards an obvious limitation of the model: the sequentiality requirement. When each ball has two choices, placing the balls sequentially is an important part of the process: the decisions of previous balls affect the placement of the next ball. If all the balls only make one random choice, however, then they can all move to their destinations simultaneously. If the speed of the algorithm is important as well as the final maximum load, this may be an important consideration.

Question 2: Is it necessary that the balls be placed sequentially in order to achieve an $O(\log\log n)$ bound on the maximum load? Can the balls be placed in parallel without degrading performance?

Our next question concerns the type of model being used. The result we have presented concerns a *static* system, where the number of balls is fixed and the process terminates after all balls have been placed. In the task-processor model, however, it seems reasonable to ask what happens if tasks enter over time and leave upon completion. Let us suppose that tasks enter at a certain rate and complete at a certain rate. We call such a system a *dynamic* system, since the process does not have a final completion point, but one is instead interested in the behavior of the system over time.[3] Azar *et al.* also study a dynamic model, but it is a *closed model*, where the number of balls remains fixed. The question of how more natural models, such as those based on queueing systems, behave has remained open.

Question 3: What if the number of balls is not fixed, but balls enter and leave the system over time?

To convince system designers and other practitioners of the potential benefit of the power of two choices, it appears necessary to be able to gauge this benefit accurately. The proof by Azar *et al.* uses various bounds and approximations which make it unclear how useful it is in predicting performance; in particular, it is not clear how large n has to be before the asymptotic result describes a noticeable effect. Although for this specific problem simulations could provide such information, it would also be more effective to have a general approach for other similar models as well.

Question 4: Is there a way to accurately predict the actual performance of GREEDY (and other strategies), instead of providing loose asymptotic bounds?

A final question, which in some sense subsumes the others, is how one should go about analyzing these types of load balancing problems. Our goal is to develop new tools and methods that can be applied not just to the specific problem at hand, but to future questions that arise in this area.

Question 5: What tools and methods can one use in studying these kinds of load balancing systems?

---

[3] The static/dynamic distinction has been made before in the routing literature; we adopt it here for scheduling.

1.1.4 Theory and practice

As is perhaps clear from the above questions, we are motivated to develop not only interesting theoretical results, but also tools and techniques for practitioners. Although the analyses provided in this thesis often utilize idealized models and our results are developed as theorems and proofs, we are guided by the principle that this underlying idea of two choices is more than an amusing mathematical trick, but instead a potentially important consideration for designers of distributed computing systems. This principle affects our approach in several ways, some of which are worth explicitly mentioning here. First, we avoid excessively esoteric models and instead focus on those that appear applicable to real situations. By contrast, some approaches to distributed models have emphasized mathematical elegance over practicality. This type of technically challenging analysis is certainly interesting in its own right; however, such results are not the goal of this work. Second, we shall attempt to provide numerical results and comparisons with simulations whenever possible. While such results may not have much theoretical interest, they dramatically impact the significance of the results to practitioners. Third, we highlight the mathematical tools that we use in the analysis, with the hope that by developing these tools appropriately, we can make them accessible to theoreticians and non-theoreticians alike.

1.1.5 Thesis outline

The remainder of the thesis is structured as follows: in Section 1.2 of the present introduction, we briefly present a proof of the result of Azar *et al.* using the original proof in [11] as our guide. This result has been the primary inspiration for this thesis, and its proof will be necessary as a point of comparison for both the results and the techniques we present here.

Chapter 2 considers the sequentiality restriction, and what happens if we try to remove the restriction from the static load balancing problem. We introduce a simple model of parallel communications, based on *rounds* of messages passed between the balls and bins. The goal of this chapter is to determine a tradeoff between the load balance and the number of rounds of communication. Surprisingly, we find that the $O(\log \log n)$ upper bound on the maximum load no longer applies if we restrict algorithms to a constant number of communication rounds. We provide both lower bounds for this model and practical algorithms that achieve these bounds, up to a constant factor. The work described in this chapter was done in conjunction with Micah Adler, Soumen Chakrabarti, and Lars Rasmussen and appeared in [6] in preliminary form.

In Chapter 3 we introduce a dynamic variant of the load balancing problem analogous to similar models from queueing theory. We then analyze this dynamic variant, using a new technique based on comparison with *infinite systems*, or systems that are arbitrarily large. The primary purpose of this section is to develop new methods for analyzing load balancing problems and to demonstrate the application of these methods to one specific problem. The dynamic model we consider, however, is interesting in its own right. This model corresponds to a scenario where tasks enter and leave a system of several processors over time at given rates. Besides the maximum load on any processor, it is also interesting to study the expected time a task spends in the system. We show that this dynamic model exhibits many of the same properties as the static model; in particular, both the maximum load and the expected time improve exponentially when incoming customers have two choices for their destination. A preliminary version of this work has appeared in [61].

In Chapter 4 we use the tools from Chapter 3 to study a host of other load balancing models, both static and dynamic. The purpose of this section is to demonstrate that our methodology has several features that give it an advantage over previous techniques: generality, accuracy, and simplicity. For example, we show how to obtain accurate predictions of the performance of the GREEDY strategy, and then compare our predictions with simulations. We follow the same procedure with several other realistic randomized load balancing schemes.

1.1.6 Previous work

Although we strive to mention the relevant work in context throughout, we provide here a brief overview of the previous work done in the area. On-line static load balancing against a worst-case adversary has been the subject of a great deal of work [9, 12, 13]; however, in the worst case competitive ratios are extremely high, suggesting that a probabilistic analysis may lead to more insight. The idea of using two (or more) hash functions to achieve better load balancing was introduced by Karp, Luby, and Meyer auf der Heide in a paper on PRAM simulation [42]. They demonstrate $O(\log\log n)$ bounds by using a universal class of hash functions, showing not only that two hash functions provide an exponential improvement over a single hash function, but also that complete independence is unnecessary. The more detailed analysis of the GREEDY strategy presented by Azar *et al.* [11] extended this work in several directions. A later paper by Broder *et al.* considers a related problem, where edges of a tree arrive randomly and must orient themselves towards a vertex [22]. Dietzfelbinger and Meyer auf der Heide [28], Goldberg *et al.* [34], MacKenzie *et al.* [55], and Stemann [72] have all developed further PRAM simulation algorithms using similar ideas. Most of these utilize collision-based protocols: if too many balls land in a bin, all of them must be rethrown.

Dynamic variations of the problem have been much less well understood. Azar *et al.* [11] do consider a dynamic variant of the problem, but it appears much less realistic than the natural queueing model we propose. A different dynamic load balancing problem, where tasks remain permanently, was considered by Atjai *et al.* in [8]. In the queueing theory literature, Schwartz [69] and Green [36] have examined parallel systems where tasks have a type that determines which servers can used, with limited success in special cases. Researchers have placed more emphasis on the model where incoming customers choose the shortest queue. Even the case where the system has just two queues is difficult; however, the problem of determining the limiting equilibrium distribution seems to have been settled for all practical purposes by Adan *et al.* using what they call the "compensation approach" [4]. For the two queue case, Adan *et al.* also settle the asymmetric case [5]. For more than two queues, Adan, van Houtum, and van der Wal derive upper and lower bounds based on comparisons with related systems [3]. The above articles, and the book by Adan discussing the compensation approach [2], provide a history of the shortest queue problem and some of its variants.

Distributed load balancing strategies based on information about only a limited number of other processors have been studied by Eager *et al.* [29, 30, 31] and Zhou [77]. In fact, the work of Eager *et al.* examines load balancing strategies similar to many of those we examine, and they also use an approach based on Markov chains for their analysis [29, 30, 31]. However, in order to perform their analysis, the authors assume that the state of each queue is stochastically independent of the state of any other queue (for example, see [29, p. 665]). The authors also claim, without justification, that this approach is exact in the asymptotic limit as the number of queues grows to infinity. Our work substantially improves upon their work by avoiding these assumptions, as well as by introducing several new directions in the analysis of these systems. Incidentally, as a byproduct of our work, one can also see that their claim about the asymptotic limit is correct. Zhou's work examines the effectiveness of the load balancing strategies proposed by Eager *et al.* as well as others in practice using a trace-driven simulation. Both Eager *et al.* and Zhou suggest that simple randomized load balancing schemes, based on choosing from a small subset of processors, appear to perform extremely well.

Our analysis of dynamic systems will be based on viewing the associated queueing problems as *density dependent jump Markov processes*. These processes have an interesting behavior: as the size of the system (measured by the number of queues) grows to infinity, the limiting behavior can be described by a deterministic process. Our treatment of density dependent jump Markov processes is based on the work of Kurtz [32, 49, 50, 51, 52]; more modern treatments of the underlying theory are given by Dembo and Zeitouni [27] and Shwartz and Weiss [70]. Kurtz's work has been applied to matching problems on random graphs [38, 43, 44] as well as to some queueing models [70]; here, we apply it for the first time to load balancing problems.

1.2 The GREEDY algorithm

We begin by presenting the main result and proof of Azar *et al.* from [11]. As previously mentioned, this result has been the primary inspiration for our work, and we shall refer back to the proof at several points. The proof and notation we present here is extremely close to the original; we have made some minor changes in the interest of clarity. Any mistakes or unclear statements introduced by the changes are of course our own.

We introduce some notation. In an (m, n, d) problem, m balls are placed into n initially empty bins sequentially, with the possible destinations of each ball being given by d choices made independently and uniformly at random (with replacement). The GREEDY algorithm places each ball in the least loaded of its d chosen bins at every step, with ties broken arbitrarily.

Theorem 1.1 *The maximum load achieved by* GREEDY *on a random* (n, n, d)*-problem is less than* $\frac{\log\log n}{\log d} + O(1)$ *with high probability.*

Before presenting the proof, which is somewhat technical, we briefly sketch an intuitive analysis. For any given i, instead of trying to determine the number of bins with load *exactly* i, it will be easier to study the number of bins with load *at least* i. The argument proceeds via what is, for the most part, a straightforward induction. Let the *height* of a ball be one more than the number of balls already in the bin in which the ball is placed. That is, if we think of balls as being stacked in the bin by order of arrival, the height of a ball is its position in the stack. Suppose we know that the number of bins with load at least i, over the entire course of the process, is bounded above by β_i. We wish to find a β_{i+1} such that, with high probability, the number of bins with load at least $i+1$ is bounded above by β_{i+1} over the course of the entire process with high probability. We find an appropriate β_{i+1} by bounding the number of balls of height at least $i+1$, which gives a bound for the number of bins with at least $i+1$ balls.

A ball will have height at least $i+1$ only if, for each of the d times it chooses a random bin, it chooses one with load at least i. Conditioned on the value of β_i, the probability that each choice finds a bin of load at least i is $\frac{\beta_i}{n}$. Therefore the probability that a ball thrown any time during the process joins a bin already containing i or more balls is at most $\left(\frac{\beta_i}{n}\right)^d$. For $d \geq 2$, we can conclude that the sequence β_i/n drops at least quadratically at each step in the following manner. The number of balls with height $i+1$ or more is stochastically dominated by a Bernoulli random variable, corresponding to the number of heads with n (the number of balls) flips, with the probability of a head being $\left(\frac{\beta_i}{n}\right)^d$ (the probability of being placed in a bin with i or more customers). We can find an appropriate β_{i+1} using standard bounds on Bernoulli trials, yielding $\beta_{i+1} \leq cn\left(\frac{\beta_i}{n}\right)^d$ for some constant c. The fraction $\frac{\beta_i}{n}$ therefore drops at least quadratically at each step, so that after only $j = O(\log\log n)$ steps the fraction drops below $1/n$, and we may conclude that $\beta_j < 1$. The proof is technically challenging primarily because one must handle the conditioning appropriately.

We shall use the following notation: the state at time t refers to the state of the system immediately after the tth ball is placed. $B(n,p)$ is a Bernoulli random variable with parameters n and p. The variable h_t is the height of the tth ball, and $\nu_i(t)$ and $\mu_i(t)$ refer to the number of bins with load at least i and the number of balls with height at least i at time t, respectively. We use ν_i and μ_i for $\nu_i(n)$ and $\mu_i(n)$ when the meaning is clear.

In preparation for the detailed proof, we make note of two elementary lemmas. The first statement can be proven by standard coupling methods:

Lemma 1.2 *Let $X_1, X_2, \ldots, X_n$ be a sequence of random variables in an arbitrary domain, and let $Y_1, Y_2, \ldots, Y_n$ be a sequence of binary random variables, with the property that $Y_i =$*

$Y_i(X_1, \ldots, X_{i-1})$. *If*

$$\mathbf{Pr}(Y_i = 1 \mid X_1, \ldots, X_{i-1}) \leq p,$$

then

$$\mathbf{Pr}(\sum_{i=1}^{n} Y_i \geq k) \leq \mathbf{Pr}(B(n,p) \geq k);$$

and similarly, if

$$\mathbf{Pr}(Y_i = 1 \mid X_1, \ldots, X_{i-1}) \geq p,$$

then

$$\mathbf{Pr}(\sum_{i=1}^{n} Y_i \leq k) \leq \mathbf{Pr}(B(n,p) \leq k).$$

■

The second lemma presents some useful Chernoff-type bounds that will be used frequently throughout the thesis; proofs may be found in [37].

Lemma 1.3 *If* X_i $(1 \leq i \leq n)$ *are independent binary random variables,* $\mathbf{Pr}[X_i = 1] = p$, *then the following hold:*

$$\text{For } t \geq np, \qquad \mathbf{Pr}\left(\sum_{i=1}^{n} X_i \geq t\right) \leq \left(\frac{np}{t}\right)^t \mathrm{e}^{t-np}. \tag{1.1}$$

$$\text{For } t \leq np, \qquad \mathbf{Pr}\left(\sum_{i=1}^{n} X_i \leq t\right) \leq \left(\frac{np}{t}\right)^t \mathrm{e}^{t-np}. \tag{1.2}$$

In particular, we have

$$\mathbf{Pr}\left(\sum_i X_{i=1}^n \geq \mathrm{e}np\right) \leq \mathrm{e}^{-np}, \quad \textit{and} \tag{1.3}$$

$$\mathbf{Pr}\left(\sum_i X_{i=1}^n \leq np/\mathrm{e}\right) \leq \mathrm{e}^{(2/\mathrm{e}-1)np}. \tag{1.4}$$

■

Proof of Theorem 1.1: Following the sketch earlier, we shall construct values β_i so that $\nu_i(n) \leq \beta_i$ for all i with high probability. Let $\beta_6 = \frac{n}{2\mathrm{e}}$, and $\beta_{i+1} = \frac{\mathrm{e}\beta_i^d}{n^{d-1}}$ for $i \leq 6 \leq i^*$, where i^* is to be determined. We let $\mathcal{E}_i$ be the event that $\nu_i(n) \leq \beta_i$. Note that $\mathcal{E}_6$ holds with certainty. We now show that, with high probability, if $\mathcal{E}_i$ holds then $\mathcal{E}_{i+1}$ holds for $6 \le i \le i^* - 1$.

Fix a value of i in the given range. Let Y_t be a binary random variable such that

$$Y_t = 1 \text{ iff } h_t \ge i+1 \text{ and } \nu_i(t-1) \le \beta_i.$$

That is, Y_t is 1 if the height of the tth ball is at least $i+1$ and at time $t-1$ there are fewer than β_i bins with load at least i.

Let ω_j represent the bins selected by the j'th ball. Then

$$\mathbf{Pr}(Y_t = 1 \mid \omega_1, \ldots, \omega_{t-1}) \le \frac{\beta_i^d}{n^d} = p_i.$$

Thus, from Lemma 1.2, we may conclude that

$$\mathbf{Pr}(\textstyle\sum_{i=1}^{n} Y_t \ge k) \le \mathbf{Pr}(B(n, p_i) \ge k).$$

Conditioned on $\mathcal{E}_i$, we have $\sum Y_t = \mu_{i+1}$. Thus

$$\begin{aligned}
\mathbf{Pr}(\nu_{i+1} \ge k \mid \mathcal{E}_i) &\le \mathbf{Pr}(\mu_{i+1} \ge k \mid \mathcal{E}_i)\\
&= \mathbf{Pr}(\textstyle\sum Y_t \ge k \mid \mathcal{E}_i)\\
&\le \frac{\mathbf{Pr}(\sum Y_t \ge k)}{\mathbf{Pr}(\mathcal{E}_i)}\\
&\le \frac{\mathbf{Pr}(B(n, p_i) \ge k)}{\mathbf{Pr}(\mathcal{E}_i)}
\end{aligned}$$

We bound the tail of the binomial distribution using the formula (1.3). Letting $k = \beta_{i+1}$ in the above, we have

$$\mathbf{Pr}(\nu_{i+1} \ge \beta_{i+1} \mid \mathcal{E}_i) \le \frac{1}{\mathrm{e}^{p_i n} \mathbf{Pr}(\mathcal{E}_i)},$$

or

$$\mathbf{Pr}(\neg \mathcal{E}_{i+1} \mid \mathcal{E}_i) \le \frac{1}{n^2 \mathbf{Pr}(\mathcal{E}_i)}$$

whenever $p_i n \ge 2 \log n$.

Hence, whenever $p_i n \ge 2 \log n$, we have that if $\mathcal{E}_i$ holds with high probability then so does $\mathcal{E}_{i+1}$. To conclude we will need to handle the case where $p_i n \le 2 \log n$ separately – we shall show that if this is the case, then with high probability there are no balls of height at least $i+2$. Let i^* be the smallest value of i such that $\frac{\beta_i^d}{n^d} \le \frac{2 \log n}{n}$. It is easy to check inductively that $\beta_{i+6} \le n/2^{d^i}$, and hence $i^* \le \frac{\log \log n}{\log d} + O(1)$.

We have

$$\mathbf{Pr}(\nu_{i^*+1} \geq 6\log n \mid \mathcal{E}_{i^*}) \leq \frac{\mathbf{Pr}(B(n, 2\log n/n) \geq 6\log n)}{\mathbf{Pr}(\mathcal{E}_{i^*})} \leq \frac{1}{n^2\mathbf{Pr}(\mathcal{E}_{i^*})},$$

where the second inequality again follows from (1.3). Also,

$$\mathbf{Pr}(\mu_{i^*+2} \geq 1 \mid \mu_{i^*+1} \leq 6\log n) \leq \frac{\mathbf{Pr}(B(n, (6\log n/n)^d) \geq 1)}{\mathbf{Pr}(\mu_{i^*+1} \leq 6\log n)} \leq \frac{n(6\log n/n)^d}{\mathbf{Pr}(\mu_{i^*+1} \leq 6\log n)},$$

where the second inequality comes from applying the crude union bound. We remove the conditioning using the fact that

$$\mathbf{Pr}(\neg\mathcal{E}_{i+1}) \leq \mathbf{Pr}(\neg\mathcal{E}_{i+1} \mid \mathcal{E}_i)\mathbf{Pr}(\mathcal{E}_i) + \mathbf{Pr}(\neg\mathcal{E}_i),$$

and obtain that

$$\mathbf{Pr}(\mu_{i^*+2} \geq 1) \leq \frac{(6\log n)^d}{n^{d-1}} + \frac{i^*+1}{n^2} = O\left(\frac{1}{n}\right),$$

which implies that with high probability the maximum load achieved by GREEDY is less than $i^* + 2 = \log\log n/\log d + O(1)$. ■

The proof of Theorem 1.1 demonstrates a useful methodology for attacking these problems. We refer to this general method as the *iterated tail bounds* approach, since the main idea is to bound the successive tails β_k inductively.

For future reference, we summarize some of the other problems analyzed in [11] and [10]. A variation on the proof above suffices to handle general (m, n, d)-problems. A corresponding lower bound of $\log\log n/\log d - O(1)$ is presented for the case $m = n$, based on the following idea: bound the number of bins with load at least 1 after the $(n/2)$th ball, then bound the number of bins of height 2 after the $(3n/4)$th ball, etc. Azar *et al.* also study a dynamic model, described as follows: initially, n bins contain n balls in some arbitrary configuration. At each step, a random ball is removed and replaced into the system into one of d bins chosen independently and uniformly at random. The GREEDY strategy places the ball into the least loaded of the d bins. Azar *et al.* show that within $O(n^2 \log\log n)$ steps, the maximum load is $O(\log\log n)$ with high probability. All of these results are proved by using the iterated tail bounds approach.

Chapter 2

Parallel randomized load balancing

2.1 Introduction

In this chapter, we shall consider the question of whether the sequentiality requirement of the original result of Azar *et al.* is necessary. Although part of our interest in this problem derives from mathematical curiosity, there are interesting practical reasons for considering parallel versions of their scheme. For example, recall the task-processor model described in Section 1.1.2, where we use the GREEDY scheme to divide tasks among processors in a distributed setting. It may happen that several tasks become ready for distribution at approximately the same time. Using the sequential variation of GREEDY, only one task can be placed at a time, and hence the final task is placed only after all others; this can result in a large latency between the time the task is ready and the time it is actually placed at a processor. Avoiding this latency requires parallel placement.

To consider parallelizations of the GREEDY strategy, we will begin by introducing a model of communication: communication between balls and bins will take place over a number of *rounds*. We first show lower bounds that hold for a wide class of load balancing strategies, including natural parallelizations of the method of Azar *et al.* Our lower bound shows that, for any r-round algorithm within our class, the load is at least $\Omega\left(\sqrt[r]{\frac{\log n}{\log\log n}}\right)$ with at least constant probability, and hence no algorithm can achieve a maximum load of $O(\log\log n)$ with high probability in a fixed constant number of rounds. We then demonstrate a parallelization of GREEDY for two communication rounds that matches the lower bounds to within a constant factor, and we examine alternative parallelizations of GREEDY that are effective when the number of communication rounds is approximately equal to the maximum load. We also examine an alternative strategy used in [42] (and often used in practice) based on setting a threshold at each bin: balls that attempt to enter a bin that has already accepted a number of balls equal to its threshold in that round must be rethrown. This strategy matches the lower bounds up to a constant factor for any constant number of rounds. Our results show that thresholding strategies can achieve a useful tradeoff between communication cost and the maximum load achieved. We conclude by presenting simulation results verifying our theoretical work.

Besides the connections to the work of Karp, Luby, and Meyer auf der Heide [42] and of Azar *et al.* [11], our work is related to other work in the area of contention resolution. For example, MacKenzie, Plaxton, and Rajaraman [55], extending previous work by Dietzfelbinger and Meyer auf der Heide [28], examine contention resolution models based on the c-collision crossbar: if more than c items attempt to access a resource, none get through. Following our original work in this area, Volker Stemann [73] has developed collision-based protocols that match our lower bound over the entire range of r. He also considers extensions where n players have τ balls to distribute into the bins, and wish to do so in a way that minimizes both the time to distribute all the balls and the maximum time to distribute each ball.

2.1.1 The model

We first describe our model in terms of balls and bins. Each of m balls is to be placed in one of n bins. Each ball begins by choosing d bins as prospective destinations, each choice being made independently and uniformly at random with replacement from all possible bins. The balls decide on their final destinations using r rounds of communication, where each round consists of two stages. In the first stage each ball is able to send, in parallel, messages to any of its prospective destination bins, and in the second stage each bin is able to send, in parallel, messages to any ball from which it has ever received a message. In the final round, the balls commit to one of the prospective bins and the process terminates. Although our lower bounds will hold more generally, for our upper bounds we will constrain our messages to be of size polylog(n, m). This restriction follows from practical considerations; messages should be short, consisting of an index number or other easily handled information. The goal is to minimize the maximum load, which is defined to be the maximum number of balls in any bin upon completion.

This model is motivated by the following realistic scenario: modern computer networks often have decentralized compute-servers (bins) and client workstations issuing tasks (balls). A distributed load-balancing strategy has to assign tasks to servers. Clients are ignorant of the intention of other clients to submit tasks; contention is known only from server load. Servers are ignorant of tasks from clients that have not communicated with them. It is also prohibitively expensive for clients to globally coordinate task submissions. The primary objectives are to minimize the maximum load achieved as well as the number of communication rounds required. Reducing the number of rounds is an important goal since, in a network setting, the time to complete a round is determined by network latency, which is generally orders of magnitude higher than CPU cycle times.

We will examine a class of simple strategies that include many of the standard algorithms presented in the literature [11, 42, 55, 72, 73]. The strategies we restrict our attention to are *non-adaptive*, in that the possible destinations are chosen before any communication takes place. We will also restrict our discussion to strategies that are *symmetric*, in the sense that all balls and bins perform the same underlying algorithm and all possible destinations are chosen independently and uniformly at random. We believe that these restrictions have practical merit, as an algorithm with these properties would be easier to implement and modify even as the underlying system changes.

Informally, we shall say that an algorithm functions *asynchronously* if a ball (or bin) has to wait only for messages addressed to it (as opposed to messages destined elsewhere). That is, balls and bins are not required to wait for a round to complete before continuing. An algorithm requires *synchronous* rounds if there must exist a synchronization barrier between some pair of rounds; that is, a ball or bin must explicitly wait for an entire previous round to complete before sending a message. In many distributed settings, the ability of an algorithm to function asynchronously can be a significant advantage; an algorithm with synchronous rounds needs some notion of global time to maintain coordination. Note that the algorithm of Azar *et al.* achieves final load no worse than $O(\log\log n)$, but requires $\Omega(n)$ synchronous rounds. Also, the obvious strategy of having the balls choose random I.D. numbers and applying standard sorting methods requires much more sophisticated centralized communication.

2.2 Lower bounds

2.2.1 The random graphs model

We first develop a general model for lower bounds that captures a class of non-adaptive, symmetric load balancing strategies. Our lower bounds are expressed in terms of the number of rounds of communication, r, and the number of choices available to each ball, d. In Section 2.2.2, we will focus on the case where $d = 2$ and $r = 2$, extending the results to arbitrary values of r and d in Section 2.2.3.

For our bounds, we will rephrase the balls and bins problem in terms of a random graph orientation problem. The relationship between balls and bins problems and random graphs has been noted previously [8, 42, 55]; we thank Claire Kenyon and Orli Waarts for suggesting this model and several helpful related ideas. Here, we show that proving a lower bound for the balls and bins problem is equivalent to showing that, with sufficiently high probability, a specific subgraph appears in a random graph. These results on random graphs may be of independent interest.

We temporarily restrict ourselves to the case of $d = 2$. Associate with each bin a vertex of a graph with n vertices. Each ball can be represented by an undirected edge in this graph, where the vertices of the edge correspond to the two bins chosen by the ball. (For convenience, in this section, we assume that each ball chooses two bins *without* replacement. This has the effect of ensuring that no self-loops arise in the graph. Multiple edges, however, may arise: these correspond to two balls that have chosen the same pair of bins. Our proofs may be modified to allow self-loops, and our restriction does not change our asymptotic results.) With each edge we shall associate an orientation. An edge, or ball, shall be oriented toward the vertex, or bin, that it chooses as its final destination. The goal of the algorithm is thus to minimize the maximum indegree over all vertices of the graph. In the case where there are m balls and n bins, the corresponding graph is a random graph from the set of all graphs with n vertices and m edges, where an edge may occur with multiplicity greater than one. We shall focus on the case $m = n$, since this is the most interesting case in terms of behavior.

We now characterize communication in this model. For each round of communication, every ball and bin will determine a larger portion of the graph around it. Following standard terminology, we define the *neighborhood* of an edge e, denoted by $N(e)$, to be the set of all edges incident to an endpoint of e. For a set S of edges, we write $N(S)$ for

$\cup_{e \in S} N(e)$. The neighborhood of a vertex v, denoted by $N(v)$, is the set of all edges incident to v. We shall also make use of the following definitions:

Definition 2.1 *The* l-neighborhood *of an edge* e*, denoted by* $N_l(e)$*, is defined inductively by:* $N_1(e) = N(e)$, $N_l(e) = N(N_{l-1}(e))$.

Definition 2.2 *The* (l, x)-neighborhood *of an edge* $e = (x, y)$*, denoted by* $N_{l,x}(e)$*, is defined inductively by:* $N_{1,x}(e) = N(x) - \{e\}$, $N_{l,x}(e) = N(N_{l-1,x}(e)) - \{e\}$.

Intuitively, for each round of communication, a ball learns about a larger neighborhood in the graph. Specifically, since we are working towards lower bounds, we may assume that the messages sent by the bins contain all available information whenever possible. Consider an r round protocol for the balls and bins problem where balls commit to their final choice in the rth round. In this case, we may assume a ball knows everything about the balls in its $(r-1)$-neighborhood, and no more, before it must commit to a bin in the rth round; this may be verified formally by a simple induction argument.

We now describe an assumption that we use to show that the final load is high with constant probability. The l-neighborhood of a ball $e = (x, y)$ splits into two subgraphs corresponding to $N_{l,x}(e)$ and $N_{l,y}(e)$; these are the parts of the neighborhood the ball learns about from each bin. Suppose that these two subgraphs of the ball's l-neighborhood are isomorphic rooted trees, with the roots being x and y. In this case we say that the ball has a *symmetric* l*-neighborhood*, or, more graphically, we say that the ball is *confused* (see Figure 2.1). The ball has no reason to prefer one bin over another, and must essentially choose randomly. For the moment, we explicitly assume that in this situation the ball chooses a bin randomly with probability $\frac{1}{2}$; we shall expand on this shortly.

Assumption 2.3 *If a ball has a symmetric* $(r-1)$*-neighborhood, then in any protocol of* r *rounds it chooses a destination bin with a fair coin flip.*

2.2.2 The $d = 2$, $r = 2$ case

If many confused balls are incident on one bin, with constant probability, over half of them will opt for the bin, which will become overloaded. We show that for r and T suitably related to n, a random graph G with n vertices and n edges has, with high probability, an *isolated* (T, r)-tree, defined as follows:

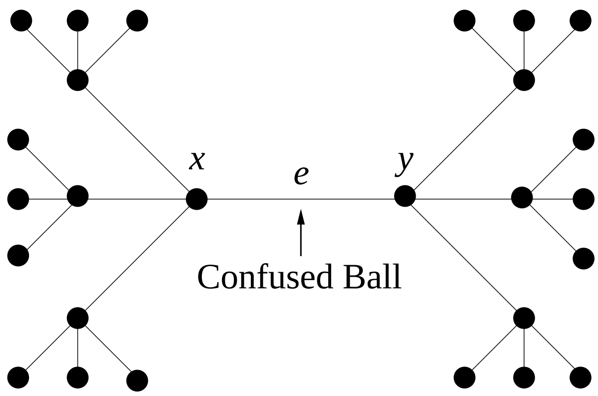

Figure 2.1: The central edge corresponds to a confused ball– its left and right neighborhoods ($N_{2,x}(e)$ and $N_{2,y}(e)$) appear the same.

Definition 2.4 *A* (T, r) tree *is a rooted, balanced tree of depth* r*, such that the root has degree* T *and each internal node has* $T-1$ *children. A* (T, r) *tree is* isolated *in a graph* G *if it is a connected component of* G *with no edges of multiplicity greater than one.*

Note that a (T, r) tree is slightly different from a T-ary tree of depth r. (See Figure 2.2.)

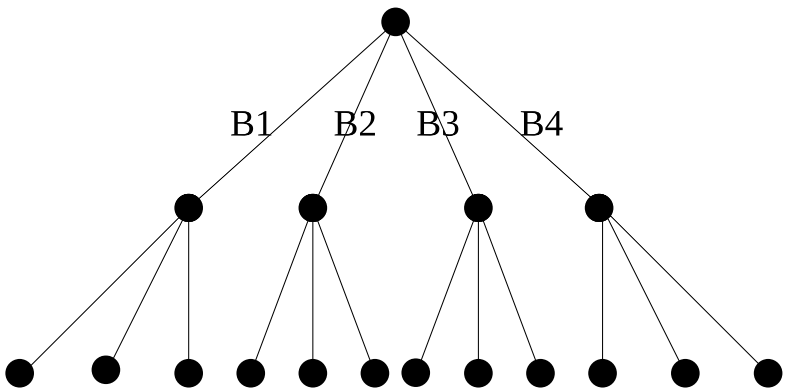

Figure 2.2: A (4,2) tree. Each vertex has degree 4, and the depth of the tree is 2. Balls B1-B4 will be confused after one just round of communication, and hence each orients itself to the root with probability 1/2.

We shall show that random graphs contain (T, r) trees of a suitable size. For convenience, we begin with the simple case of $d = 2$ and $r = 2$.

Theorem 2.5 *With constant probability, a random graph with n vertices and n edges contains an isolated $(T, 2)$ tree with $T = (\sqrt{2} - o(1))\sqrt{\frac{\log n}{\log\log n}}$.*

Since, with constant probability, half of the confused edges in an isolated $(T, 2)$ tree adjacent to the root will orient themselves toward it (by Assumption 2.3), the above theorem immediately yields the following corollary:

Corollary 2.6 *Any non-adaptive, symmetric load distribution strategy for the balls and bins problem with n balls and n bins satisfying Assumption 2.3, where $d = 2$ and $r = 2$, has a final load at least $\left(\frac{\sqrt{2}}{2} - o(1)\right)\sqrt{\frac{\log n}{\log\log n}}$ with at least constant probability.*

Corollary 2.6 demonstrates that the $O(\log\log n)$ bounds achieved by Azar *et al.* using the GREEDY strategy cannot be achieved by any two round strategy where each ball has two choices.

Although we prove Theorem 2.5 for the case where m, the number of edges, and n, the number of bins, are equal, it will be useful to write the number of edges as m throughout the proof. Besides making the proof clearer, this will allow us to extend the theorem easily to a broader range of m; this is discussed after the proof.

Proof of Theorem 2.5: Let $\vec{v} = (v_0, v_1, \ldots, v_{T^2})$ be a vector of $T^2 + 1$ vertices. Let $X_{\vec{v}}$ be an indicator random variable that is 1 if v_0 is the root of an isolated $(T, 2)$ tree, $v_1, \ldots, v_T$ are the nodes of depth 1, $v_{T+1}, \ldots, v_{2T-1}$ are the children of v_1, and so on, and let $X = \sum_{\vec{v}} X_{\vec{v}}$. We show that $X > 0$ with at least constant probability by determining the expectation and variance of X and applying the simple bound (from [21], equation (3) of I.1):

$$\Pr(X = 0) \leq 1 - \frac{\mathsf{E}[X]^2}{\mathsf{E}[X^2]}.$$

The multinomial coefficient $\binom{n}{1;T;T-1;\ldots;T-1}$, where the sum of the lower terms $1 + T + (T-1) + \ldots + (T-1)$ is $T^2 + 1$, gives the number of possible choices for $\vec{v}$; we must first choose the root, and then the T children of the root, and then the $T-1$ children for each child. We now choose a specific $\vec{v}$ and determine the probability p that $X_{\vec{v}}$ is 1. If $X_{\vec{v}}$ is 1, there must be T^2 edges corresponding to the (T, r) tree connecting the vertices of $\vec{v}$ and no other edges incident to these vertices. We must first choose T^2 of the m edges to make up the tree: there are $\binom{m}{T^2}(T^2)!$ ways of doing this. The remaining edges must lie on the remaining $n-(T^2+1)$ vertices so that the tree is isolated. Hence

$$p = \frac{\binom{n-(T^2+1)}{2}^{m-T^2}\binom{m}{T^2}(T^2)!}{\binom{n}{2}^m}.$$

Using the linearity of expectations, we have

$$\mathsf{E}[X] \quad = \quad \frac{\binom{n}{1;T;T-1;\ldots;T-1}\binom{n-(T^2+1)}{2}^{m-T^2}\binom{m}{T^2}(T^2)!}{\binom{n}{2}^m}.$$

This unwieldy expression can be simplified by canceling appropriately and noting that we will choose T small enough so that many terms are $o(1)$ (in the case $m=n$). For instance, if $T=o(\log n)$, then we have

$$\frac{\binom{n-(T^2+1)}{2}^{m-T^2}}{\binom{n}{2}^{m-T^2}} \quad = \quad \mathrm{e}^{-2(T^2+1)m/n}(1+o(1))\,; \tag{2.1}$$

$$\frac{\binom{m}{T^2}(T^2)!}{n^{T^2}} \quad = \quad \left(\frac{m}{n}\right)^{T^2}(1+o(1))\,; \tag{2.2}$$

$$\frac{\binom{n}{1;T;T-1;\ldots;T-1}}{(n-1)^{T^2}} \quad = \quad \frac{n}{((T-1)!)^{T+1}T}(1+o(1))\,;. \tag{2.3}$$

We thus obtain

$$\mathsf{E}[X] \quad = \quad \frac{n\left(\frac{2m}{n}\right)^{T^2}}{\mathrm{e}^{2(T^2+1)m/n}((T-1)!)^{T+1}T}(1+o(1)). \tag{2.4}$$

We now examine how to compute $\mathsf{E}[X^2]$. Note that, because we are considering only isolated $(T,2)$ trees, if $\vec{v}\neq\vec{w}$ then $X_{\vec{v}}$ and $X_{\vec{w}}$ can both equal 1 if and only if $\vec{v}$ and $\vec{w}$ consist of disjoint sets of vertices or are equal. This simplifies the calculation of $\mathsf{E}[X^2]$ considerably. Since

$$\mathsf{E}[X^2] \quad = \quad \mathsf{E}[X] + \sum_{\vec{v}\neq\vec{w}} \mathsf{E}[X_{\vec{v}}X_{\vec{w}}],$$

it suffices to compute the second term. The calculation is similar to that for $\mathsf{E}[X]$.

The number of possible disjoint pairs $\vec{v}$ and $\vec{w}$ is $\binom{n}{1;T;T-1;\ldots;T-1;1;T;T-1;\ldots;T-1}$, and the probability q that a given disjoint pair $\vec{v}$ and $\vec{w}$ yields two isolated $(T,2)$ trees is

$$q = \frac{\binom{n-(2T^2+2)}{2}^{m-2T^2}\binom{m}{2T^2}(2T^2)!}{\binom{n}{2}^m}.$$

From these terms we derive $\sum_{\vec{v} \neq \vec{w}} \mathsf{E}[X_v X_w] = \mathsf{E}[X]^2(1+o(1))$, by simplifying with equations entirely similar to equations (2.1), (2.2), and (2.3). We thus have that $\mathsf{E}[X^2] = \mathsf{E}[X] + \mathsf{E}[X]^2(1+o(1))$. It now suffices to choose a T such that $\mathsf{E}[X]$ is bounded below by a constant. Taking the logarithm of both sides of Equation 2.4, we find this will hold as long as

$$T^2 \log T + \frac{2T^2 m}{n} - T^2 \log \frac{2m}{n} \leq \log n + o(\log n). \tag{2.5}$$

Hence for $m = n$ there exists a $(T, 2)$ tree with $T = (\sqrt{2} - o(1))\sqrt{\frac{\log n}{\log \log n}}$ with constant probability. ■

Remark: Although we have stated Theorem 2.5 only for the case $m = n$, it is clear that nearly the same proof, and hence equation (2.5), applies for a range of m. For example, if $m = \frac{n}{\log^k n}$ for some fixed k, then we again have $\Omega\left(\sqrt{\frac{\log n}{\log \log n}}\right)$ bounds on the maximum load with at least constant probability. The important points to check are where we have stated that some terms go to $o(1)$, as in equations (2.1), (2.2), and (2.3), which places restrictions on the possible values of m and T. It is also worth noting that equation (2.5) can be improved somewhat. We have insisted up to this point that our trees be isolated, even though there is no reason that the leaf nodes cannot be adjacent to nodes outside the tree. Taking advantage of this fact would reduce the $\frac{2T^2 m}{n}$ terms of equation (2.5) to $\frac{2Tm}{n}$. Although this does not affect the bound when $m = n$ in this case, it will be important in the generalization we consider in the following section. ■

Although it may at first seem unreasonable to insist that balls with symmetric r-neighborhoods choose a bin randomly, obvious tie-breaking schemes do not affect the lower bound. For instance, if the balls are ordered at the bins, either by random I.D. numbers or by a random permutation, and then choose a bin according to their rank, the balls are essentially choosing a bin at random. The proof can also easily be modified for the case where the balls are ranked at the bins by some fixed ordering by using the symmetry of the destination choices of the balls. Similarly, if bins are numbered and given a preferred ordering in case of ties, then with constant probability there is still a $(T, 2)$ tree whose root has the given final load.

2.2.3 The general case

One can extend Theorem 2.5 to the case where $d > 2$ and $r > 2$; in fact, the extension also applies if r and d grow sufficiently slowly with n.

When $r > 2$ and $d = 2$, the balls and bins scenario can again be reduced to a random graph problem; instead of showing the existence of a $(T, 2)$ tree, one needs to demonstrate the existence of a (T, r) tree. When $d > 2$ we must consider hypergraphs instead of graphs. In this model, balls correspond to hyperedges of d distinct vertices in the hypergraph. The degree of a vertex is the number of incident hyperedges. A tree of hyperedges is simply a connected acyclic hypergraph, and the depth of a tree is the number of hyperedges in a longest path from the root to a leaf.

Definition 2.7 *A* (T, r, d) tree *is a rooted, balanced tree of depth r with edges of size d, such that the root has degree T and each internal node has $T - 1$ children. A (T, r, d) tree is* near-isolated *in a graph G if it has no edges of multiplicity greater than one and no edge of G other than the tree edges are incident to any non-leaf node of the tree.*

Figure 2.3 gives an example of a $(3, 2, 3)$ tree. We have also defined the notion of a near-isolated tree, since, as suggested in the remark after Theorem 2.5, by considering near-isolated trees we will be able to tighten our bounds in the general case.

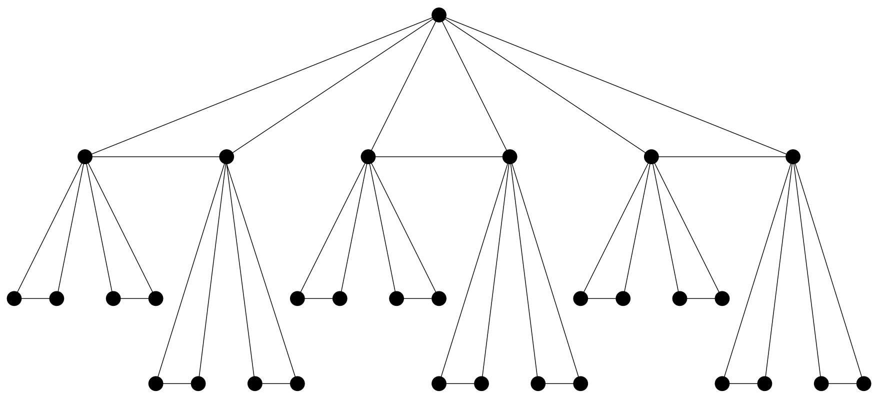

Figure 2.3: A (3,2,3) tree. Each triangle corresponds to a hyperedge of three vertices.

The l-neighborhood and (l, x)-neighborhood of a ball can be defined for hypergraphs similar to Definitions 2.1 and 2.2. As in Assumption 2.3, we will assume that if a ball has a symmetric $(r-1)$-neighborhood, it chooses one of the d bins uniformly at random at the end of an r round algorithm; for convenience, we still call this Assumption 2.3. Thus the root of a near-isolated (T,r,d) tree will end with $\frac{T}{d}$ balls with at least constant probability. As we shall see, whether a near-isolated (T,r,d) tree exists is essentially a matter of its size, in terms of the number of edges it contains.

In the remainder of this section, we shall prove the following theorem:

Theorem 2.8 *For any fixed r and d, there exists a $T = \Omega\left(\sqrt[r]{\frac{\log n}{\log\log n}}\right)$ such that with constant probability, a random graph with n vertices and n edges of size d contains a near-isolated (T,r,d) tree.*

The theorem immediately yields the following corollary:

Corollary 2.9 *Any non-adaptive, symmetric load distribution strategy for the balls and bins problem satisfying Assumption 2.3 where d and r are constants and $m = n$ has a final load at least $\Omega\left(\sqrt[r]{\frac{\log n}{\log\log n}}\right)$ with constant probability.*

Corollary 2.9 generalizes Corollary 2.6 by demonstrating that the $O(\log\log n)$ bounds achieved by Azar *et al.* using the GREEDY strategy cannot be achieved by any strategy in which each ball has a fixed constant number of choices and in which only a constant number of rounds are used.

The constant factor in the lower bound (for r and d fixed) is dependent on d. The proof of the theorem also yields results when d grows with n. With constant probability the final load is $\frac{T}{d}$ if there is a (T,r,d) tree in the corresponding hypergraph. Similarly, if there is a (T,r,d) tree in the corresponding hypergraph, then with probability d^{-T} the final load is T; this can be used to give negative results by showing that no non-adaptive, symmetric load distribution strategy achieves load T with high probability when $d^T = o(n)$. For example, we have the following corollary:

Corollary 2.10 *Any non-adaptive, symmetric load distribution strategy for the balls and bins problem satisfying Assumption 2.3 with $m = n$ where $d = O\left(\frac{\log\log n}{\log\log\log n}\right)$ and $r = O\left(\frac{\log\log n}{\log\log\log n}\right)$ has a final load at least $\Omega\left(\frac{\log\log n}{\log\log\log n}\right)$ with probability at least $O\left(1/\log^c n\right)$ for some constant c (dependent on d and r).*

We warn the reader that the proof of Theorem 2.8, although not difficult, is somewhat technical, and it can be skipped on the first reading without affecting the understanding of the rest of the thesis.

Proof of Theorem 2.8: As in Theorem 2.5, although we are considering only the case $m = n$, we continue to distinguish m and n when writing the proof, in the interests of enhancing clarity and allowing generalizations to other values of m where suitable.

We begin by finding the expected number of (T, r, d) trees. Let V denote the total number of vertices in the desired (T, r, d) tree, let V_I denote the number of vertices on internal edges of the tree, and let E denote the total number of hyperedges of the tree. A given list of V vertices corresponds to a unique (T, r, d) tree in a canonical way. For a given list of V vertices, the probability p that the corresponding tree component exists is

$$p = \frac{\binom{n-V_I}{d}^{m-E}\binom{m}{E}E!}{\binom{n}{d}^m}.$$

That is, we must choose from the m edges the E edges of the tree, and all other edges cannot be incident to an internal vertex.

To compute the number of possible trees, we consider all lists of V vertices, where the first vertex corresponds to the root, the next $T(d-1)$ vertices correspond to the first T edges, and so on. Each possible re-ordering of the vertices that make up an edge leads to the same tree; also, the subtrees at each level can be permuted without changing the tree. Keeping these facts in mind, we find the total possible number of vectors corresponding to distinct near-isolated trees, which we denote by N, is

$$\begin{aligned} N &= \frac{\binom{n}{1;T(d-1);(T-1)(d-1);\ldots;(T-1)(d-1)}\binom{T(d-1)}{d-1;\ldots;d-1}\binom{(T-1)(d-1)}{d-1;\ldots;d-1}^{(E-T)/(T-1)}}{T![(T-1)!]^{(E-T)/(T-1)}} \\ &= \frac{n!}{(n-V)![(d-1)!]^E T[(T-1)!]^{(E-1)/(T-1)}}, \end{aligned}$$

where in the first multinomial coefficient, the sum of the terms on the second level, $1 + T(d-1) + (T-1)(d-1) + \ldots + (T-1)(d-1)$, is V.

Routine calculations yield that $V = 1 + T(d-1)\frac{[(T-1)(d-1)]^r - 1}{(T-1)(d-1)-1}$, $E = \frac{V-1}{d-1}$, and $V_I = 1 + T(d-1)\frac{[(T-1)(d-1)]^{r-1}-1}{(T-1)(d-1)-1}$. For suitable values of T (and hence E and V), after absorbing lower order terms the product Np reduces to:

$$Np = \frac{n\left(\frac{m}{n}\right)^E d^E(1+o(1))}{e^{V_I dm/n}T((T-1)!)^{(E-1)/(T-1)}}.$$

Hence the expected number of (T, r, d) trees can be made at least a constant for suitable choices of T, r, and d; this requires

$$\log n + E\log\frac{m}{n} + E\log d \quad\geq\quad \frac{V_I dm}{n} + (E-1)\log(T-1) + o((E-1)\log(T-1)).$$

Noting that $E \approx (Td)^r$, we find that (up to lower order terms) we require $\log n \geq (Td)^r \log T$ when $m = n$. In particular, when d is a fixed constant we can find a T of size at least $\Omega\left(\sqrt[r]{\frac{\log n}{\log\log n}}\right)$ such that the expected number of (T, r, d) trees is at least a constant when $m = n$.

We must now also show that the variance is not too large. Recall that

$$\mathsf{E}[X^2] \quad=\quad \mathsf{E}[X] + \sum_{\vec{v}\neq\vec{w}} \mathsf{E}[X_{\vec{v}}X_{\vec{w}}].$$

Finding $\sum_{\vec{v}\neq\vec{w}} \mathsf{E}[X_{\vec{v}}X_{\vec{w}}]$ is more difficult than in Theorem 2.5, because the trees are not isolated. There are two cases: $\vec{v}$ and $\vec{w}$ share no internal vertices, or they share at least one internal vertex. If $\vec{v}$ and $\vec{w}$ share no internal vertices, then the probability p that $X_{\vec{v}}$ and $X_{\vec{w}}$ are both 1 is

$$p \quad=\quad \frac{\binom{n-2V_I}{d}^{m-2E}\binom{m}{2E}2E!}{\binom{n}{d}^m}.$$

The number of pairs of disjoint $\vec{v}$ and $\vec{w}$ can be found in the following manner: we first choose the $2V_I$ internal vertices, and then for each tree we choose the remaining the $V - V_I$ vertices from the $n - 2V_I$ vertices. Hence, the number of pairs is

$$\frac{n!(n-2V_I)!(n-2V_I)!}{(n-2V_I)!(n-V_I-V)!(n-V_I-V)![(d-1)!]^{2E}T^2[(T-1)!]^{2(E-1)/(T-1)}}.$$

If $\vec{v}$ and $\vec{w}$ share an internal vertex, then the root of one tree must be the internal vertex of another. Without loss of generality let $\vec{v}$ be the tree whose root is not an internal vertex. Then we use the following argument to bound $\mathsf{E}[X_{\vec{v}}X_{\vec{w}}]$.

$$\begin{aligned}
\mathsf{E}[X_{\vec{v}}X_{\vec{w}}] &= \mathbf{Pr}(X_{\vec{v}} = 1 \text{ and } X_{\vec{w}} = 1) \\
&= \mathbf{Pr}(X_{\vec{v}} = 1)\mathbf{Pr}(X_{\vec{w}} = 1 | X_{\vec{v}} = 1) \\
&= \mathsf{E}[X_{\vec{v}}]\mathbf{Pr}(X_{\vec{w}} = 1 | X_{\vec{v}} = 1).
\end{aligned}$$

Let us now look at a specific, fixed $\vec{v}$. Let S_I be the set of internal vertices for $\vec{v}$, and let z be a fixed vertex sharing an edge with the root of $\vec{v}$. We then have the following bound:

$$\begin{aligned}\sum_{\vec{w}\neq\vec{v}} \mathsf{E}[X_{\vec{v}}X_{\vec{w}}] &= \mathsf{E}[X_{\vec{v}}]\mathbf{Pr}(X_{\vec{w}}=1|X_{\vec{v}}=1)\\ &\leq \sum_{y\in S_I} \mathsf{E}[X_{\vec{v}}]\mathbf{Pr}(y \text{ is a root of a } (T,r,d) \text{ tree } |X_{\vec{v}}=1)\\ &\leq V_I\,\mathsf{E}[X_{\vec{v}}]\mathbf{Pr}(z \text{ is a root of a } (T,r,d) \text{ tree } |X_{\vec{v}}=1).\end{aligned}$$

The last line captures the fact that the greater the overlap between $\vec{v}$ and $\vec{w}$, the more likely both trees lie in the graph.

Let $p_z = \mathbf{Pr}(z \text{ is a root of a } (T,r,d) \text{ tree } |X_{\vec{v}} = 1)$, and let E_2 be the number of new hyperedges one must add to the tree given by $\vec{v}$ so that z is also the root of a near-isolated (T,r,d) tree. We have $E_2 = \frac{[(T-1)(d-1)]^r}{d-1}$, and from this we calculate p_z to find

$$p_z = \frac{\binom{n-V_I}{(T-1)(d-1);\ldots;(T-1)(d-1)}\left[\frac{\binom{(T-1)(d-1)}{d-1;\ldots;d-1}}{(T-1)!}\right]^{E_2/(T-1)}\binom{n-V_I-(E_2/(T-1))}{d}^{m-E-E_2}\binom{m-E}{E_2}E_2!}{\binom{n-V_I}{d}^{m-E}}.$$

Or, more conveniently,

$$p_z \approx \frac{d^{E_2}\mathrm{e}^{-mdE_2/n(T-1)}(m/n)^{E_2}}{(T-1)!^{E_2/(T-1)}}.$$

It is easy to check that $p_z = o(1/V_I)$ when $m = n$. Summing over all cases yields

$$\mathsf{E}[X^2] = (\mathsf{E}[X] + \mathsf{E}[X]^2)(1 + o(1)),$$

which is sufficient to prove the theorem. ■

2.3 The Poisson approximation

We now switch from proving lower bounds to examining parallel algorithms for load balancing based on the GREEDY idea. Before developing any particular algorithms, it will be useful to derive a general tool that we will use often in what follows. The main difficulty in analyzing balls and bins problems is that it is often hard to handle the dependencies that naturally arise in such systems. For example, if one bin is empty, then it is less likely that another bin is empty; the loads of the various bins are correlated. It will be useful to have a general way to circumvent these sorts of dependencies. We show here how to do so when we are examining the probability of a rare event. This idea – finding ways around natural dependencies that frustrate analysis – is a common theme that will occur again in this thesis.

It is well known that after throwing m balls independently and uniformly at random into n bins, the distribution of the number of balls in a given bin is approximately Poisson with mean $\frac{m}{n}$. We would like to say that the joint distribution of the number of balls in *all* the bins is well approximated by assuming the load at *each* bin is an *independent* Poisson random variable with mean $\frac{m}{n}$. This would allow us to treat bin loads as independent random variables, and hence use standard techniques such as Chernoff bounds.

Suppose m balls are thrown into n bins independently and uniformly at random, and let $X_i^{(m)}$ be the number of balls in the i-th bin, where $1 \leq i \leq n$. Let $Y_1^{(m)}, \ldots, Y_n^{(m)}$ be independent Poisson random variables with mean $\frac{m}{n}$. We will omit the superscript when it is clear from the context. In this section we will derive some relations between these two sets of random variables, adapting an argument used by Gonnet [35] to determine the expected maximum number of balls in a bin. We note that the close relationship between these two models has been observed and made use of previously, for example in [23], and tighter bounds on specific problems can often be obtained with more detailed analyses, as can be seen, for example, in [16, Chapter 6] or [41]. An advantage of the approach we present is that it is quite general and easy to apply.

Theorem 2.11 *Let $f(x_1, \ldots, x_n)$ be a non-negative function. Then*

$$\mathsf{E}[f(X_1, \ldots, X_n)] \quad \leq \quad \sqrt{2\pi e m}\, \mathsf{E}[f(Y_1, \ldots, Y_n)]. \tag{2.6}$$

Further, if $\mathsf{E}[f(X_1, \ldots, X_n)]$ is either monotonically increasing or monotonically decreasing with m, then

$$\mathsf{E}[f(X_1, \ldots, X_n)] \quad \leq \quad 4\, \mathsf{E}[f(Y_1, \ldots, Y_n)]. \tag{2.7}$$

Proof: We have that

$$\begin{aligned}
\mathsf{E}[f(Y_1, \ldots, Y_n)] &= \sum_{k=0}^{\infty} \mathsf{E}\left[f(Y_1, \ldots, Y_n) \Big| \sum Y_i = k\right] \mathbf{Pr}(\sum Y_i = k) \\
&\geq \mathsf{E}\left[f(Y_1, \ldots, Y_n) \Big| \sum Y_i = m\right] \mathbf{Pr}(\sum Y_i = m) \\
&= \mathsf{E}[f(X_1, \ldots, X_n)] \mathbf{Pr}(\sum Y_i = m)
\end{aligned}$$

where the last equality follows from the fact that the joint distribution of the Y_i given $\sum Y_i = m$ is exactly that of the X_i, as can be checked by comparing the probabilities of any given set of bin loads in both cases. As $\sum Y_i$ is Poisson distributed with mean m, we now have

$$\mathsf{E}[f(Y_1, \ldots, Y_n)] \quad \geq \quad \mathsf{E}[f(X_1, \ldots, X_n)] \frac{m^m \mathrm{e}^{-m}}{m!},$$

and using Stirling's approximation now yields equation (2.6).

If $\mathsf{E}[f(X_1, \ldots, X_n)]$ increases with m, then by a similar argument we have

$$\begin{aligned}
\mathsf{E}[f(Y_1, \ldots, Y_n)] &\geq \sum_{k=m}^{\infty} \mathsf{E}\Big[f(Y_1, \ldots, Y_n) \Big| \textstyle\sum Y_i = k\Big] \mathbf{Pr}(\textstyle\sum Y_i = k) \\
&\geq \mathsf{E}\Big[f(Y_1, \ldots, Y_n) \Big| \textstyle\sum Y_i = m\Big] \mathbf{Pr}(\textstyle\sum Y_i \geq m) \\
&= \mathsf{E}[f(X_1, \ldots, X_n)] \, \mathbf{Pr}(\textstyle\sum Y_i \geq m)
\end{aligned}$$

It is easy to check that $\mathbf{Pr}(\sum Y_i \geq m)$ can be bounded below by $1/4$, and equation (2.7) follows. The case where $\mathsf{E}[f(X_1, \ldots, X_n)]$ decreases with m is similar. ∎

From this theorem, we derive a corollary that will be central to most of our proofs. Let us call the scenario in which bin loads are taken to be independent Poisson random variables with mean $\frac{m}{n}$ the *Poisson case*, and the scenario where m balls are thrown into n bins independently and uniformly at random the *exact case*. Also, let a *load based event* be an event that depends solely on the loads of the bins.

Corollary 2.12 *A load based event that takes place with probability p in the Poisson case takes place with probability at most $p\sqrt{2\pi \mathrm{e} m}$ in the exact case. If the probability of the event is monotonically increasing or decreasing with the total number of balls, then the probability of the event is at most $4p$ in the exact case.*

Proof: Let f be the indicator function of the load based event. In this case $\mathsf{E}[f]$ is just the probability that the event occurs, and the result follows immediately from Theorem 2.11. ∎

To demonstrate the utility of this corollary, we provide a simple representative example that will prove useful later.

Lemma 2.13 *Suppose $m < \frac{n}{\log n}$, and suppose m balls are thrown independently and uniformly at random into n bins. Then, with high probability, the maximum load is $\Theta(\frac{\log n}{\log \frac{n}{m}})$.*

Proof: By Corollary 2.12, since the maximum load is monotonically increasing with the number of balls, it is sufficient to prove that the bounds hold in the Poisson case. Let p_k be the probability that any particular bin contains k or more balls.

For the lower bound, note that

$$p_k \quad \geq \quad \frac{(\frac{m}{n})^k \mathrm{e}^{-m/n}}{k!},$$

as the right hand side is simply the probability that a bin has exactly k balls. The probability that no bin has k or more balls is thus at most $(1-p_k)^n \leq \mathrm{e}^{-p_k n}$, and we need to show that $\mathrm{e}^{-p_k n} \leq \frac{1}{n}$ when $k = \Omega(\frac{\log n}{\log \frac{n}{m}})$. Taking logarithms twice yields the following sufficient condition:

$$\log k! + k \log(\tfrac{n}{m}) \quad \leq \quad \log n - O(\log\log n). \tag{2.8}$$

It is now simple to check that choosing $k = \frac{a \log n}{\log \frac{n}{m}}$ for any constant $a < \frac{1}{2}$ suffices as long as $m < \frac{n}{\log n}$.

For the upper bound, note that

$$p_k \quad \leq \quad \frac{2(\frac{m}{n})^k \mathrm{e}^{-m/n}}{k!}, \tag{2.9}$$

as can be found by bounding the probability that a bin has k or more balls by a geometric series (and using just that $m < n$). It is easy to show that when $k \geq \frac{3 \log n}{\log \frac{n}{m}}$, this probability is less than $\frac{1}{n^2}$, and thus no bin contains $\frac{3\log n}{\log \frac{n}{m}}$ or more balls with probability at least $1 - O(1/n)$ in the exact case. ∎

For completeness, we also prove a weak form of the result we stated back in Chapter 1 for the case of n balls and n bins in a similar fashion:

Lemma 2.14 *Suppose n balls are thrown independently and uniformly at random into n bins. Then, with high probability, the maximum load is $\Theta(\frac{\log n}{\log\log n})$.*

Proof: By Corollary 2.12 it is sufficient to prove that the bounds hold in the Poisson case. Let p_k be the probability that any particular bin contains k or more balls.

For the lower bound, note that $p_k \geq \frac{1}{ek!}$. Now, as in Lemma 2.13, we find that $e^{-p_k n} \leq \frac{1}{n}$ when $k = \Omega(\frac{\log n}{\log\log n})$.

For the upper bound, note that $p_k \leq \frac{1}{k!}$, and hence for some k of size $O(\frac{\log n}{\log\log n})$, p_k is $O(1/n^2)$, from which the claims follows. ∎

We emphasize that Corollary 2.12 will prove useful to us because in the Poisson case all bin loads are independent. This independence allows us to use various forms of Chernoff bounds in the Poisson case, and then transfer the result to the exact case, greatly simplifying the analysis.

2.4 Parallel GREEDY

The lower bounds in Section 2.2 show that if the number of communication rounds and possible destinations for a ball are fixed, the $\frac{\log\log n}{\log d} + O(1)$ maximum load bound of [11] no longer applies. We therefore seek ways to parallelize the GREEDY strategy, with the aim of matching the lower bounds. We first deal with the case of two rounds in Section 2.4.1, and then consider multiple rounds in Section 2.4.2. For these sections, we restrict ourselves to the case $d \geq 2$.

2.4.1 A two round parallelization of GREEDY

We begin with a more formal description of GREEDY. Each ball a will at some point in the algorithm independently *choose* d destination bins $i_1(a), i_2(a), \ldots i_d(a)$ (with replacement). We may assume that these choices are made in parallel as the first step in the algorithm; this assumption makes it clear that GREEDY is non-adaptive. Next, each ball a decides, solely by communicating with $i_1(a), \ldots, i_d(a)$, to which bin it shall *commit*. Once a ball has committed to a bin, its decision cannot be reversed. We note that ties in this and other algorithms are broken arbitrarily and the d bin choices are made with replacement unless stated otherwise.

CHOOSE(d):
in parallel: each ball a
chooses u.a.r. d bins $i_1(a), \ldots, i_d(a)$

GREEDY(d):
call CHOOSE(d)
sequentially: each ball a
queries bins $i_1(a), \ldots, i_d(a)$ for current load
commits to bin with smallest load

We will break the sequentiality of GREEDY by letting the balls choose between $i_1(a), \ldots, i_d(a)$ according to the selections made by the other balls in the *initial* stage of the process. Let all the balls inform their d choices that they have been chosen by sending them each a *request*. We shall refer to the d requests as *siblings*.

Each bin then creates a list of the requests it has received. The bins may order their lists arbitrarily. However, if they handle requests in the order they arrive, the algorithm may function asynchronously. Notice that we make no claim that the requests arrive at the bins in any particular order.

The *height* of a request is its position in the request list it belongs to. The bins now send back the heights of their requests to the balls. Finally, each ball commits to the bin in which its request had the smallest height. This allows the entire process, which we call PGREEDY, to finish in only two rounds:

PGREEDY(d):
call CHOOSE(d)
in parallel: each ball a
sends requests to bins $i_1(a), \ldots, i_d(a)$
in parallel: each bin i
creates list of received requests
sends heights to requesting balls
in parallel: each ball a
commits to bin with smallest height

Note that Corollary 2.8 provides a lower bound for the PGREEDY strategy. We now prove a matching upper bound on the maximum load achieved by PGREEDY.

Theorem 2.15 *For fixed d and $m = n$, the maximum load achieved by* PGREEDY(d) *is at most* $O\left(\sqrt{\frac{\log n}{\log \log n}}\right)$ *with high probability.*

Proof: As in Theorems 2.5 and 2.8, although we are considering only the case $m = n$, we maintain the distinction between m and n when writing the proof.

The outline of the proof is as follows: consider any bin i, and consider all balls with requests of height larger than some T_1 in bin i. For such a ball to choose bin i, all of its siblings' requests must have height at least T_1, and hence they must all have landed in bins that received at least T_1 requests. By choosing T_1 large enough, we can make the probability that a request at bin i of height at least T_1 chooses bin i very small, and thereby bound the number of balls that choose bin i.

We begin with some technical details. First, there may be balls that have one or more siblings choose bin i. The expected number of such balls is $O\left(\frac{md^2}{n^2}\right)$; as $m = n$ and d is fixed, with high probability the number of such balls is bounded by a constant. We can therefore absorb these balls in the $O(1)$ term and ignore them in the remainder of the proof. Second, we choose a bound M such that with high probability, bin i receives no more than M requests. (For example, in this case, we may take $M = O(\log n)$.) Conditioned on both these high probability events occurring, the set R of requests sent to a bin other than i are distributed in the remaining $n - 1$ bins independently and uniformly.

Consider all requests in i of height at least T_1, all of whose siblings lie outside i; call this set of requests I. We prove that, with sufficiently high probability, fewer than T_2 elements of i have siblings whose heights are all T_1 or more.

Consider the subprocess of requests R arriving at the bins other than i. We can imagine these requests arriving sequentially at the bins according to some arbitrary ordering. Let time t be the instant immediately after the t'th such request arrives.

We now use an innovation from the proof of Theorem 1.1. Let $\mathcal{E}_t$ be the event that, at time t, no more than N bins have received more than T_1 requests from R, for some N to be determined later. Also, let the random variable X_t equal 1 if the height of the t'th request is greater than T_1, and 0 otherwise. Now, for $r \in I$, let $S(r)$ be the set of arrival times for the siblings of r, and let the random variable Y_r equal 1 if, for every $t \in S(r)$, $X_t = 1$ and $\mathcal{E}_t$ occurs; Y_r is 0 otherwise. That is, $Y_r = 1$ if and only if all the siblings of r are of height at least T_1, but the number of bins of height T_1 has not become higher than N before all siblings of r arrive.

We define $\mathcal{E}$ to be the event that $\mathcal{E}_t$ is true for all t. Conditioned on $\mathcal{E}$, we have that $\sum_{r \in R} Y_r$ is an upper bound on the number of balls with requests of height at least T_1 at bin i that choose bin i as their final destination. Note that $\mathbf{Pr}(Y_r = 1) \leq \left(\frac{N}{n}\right)^{d-1}$. It follows that the sum of any subset of the Y_i is stochastically dominated by the sum of the same number of independent Bernoulli variables with parameter $\left(\frac{N}{n}\right)^{d-1}$ by Lemma 1.2. Now we choose an N so that $\mathcal{E}$ is true with high probability. In the Poisson case, the probability that a bin has T_1 requests is $\frac{\mathrm{e}^{-md/n}(md/n)^{T_1}}{T_1!}$. As long as $T_1 > 2md/n$, then the probability that a bin has at least T_1 requests is at most $\frac{2\mathrm{e}^{-md/n}(md/n)^{T_1}}{T_1!}$. Applying Chernoff bounds (Lemma 1.3, equation (1.1)), with high probability the number of bins with at least T_1 requests is at most $N = \frac{4n\mathrm{e}^{-md/n}(md/n)^{T_1}}{T_1!}$ in the Poisson case. Since the number of bins with at least T_1 requests is monotonically increasing in the number of requests, the same is true in the exact case as well by Corollary 2.12.

We use the bound on N to bound the number of balls with requests of height at least T_1 in i whose siblings all have height at least T_1. Again, using Chernoff bounds (Lemma 1.3, equation (1.1)), we have

$$\mathbf{Pr}\left[\sum_{r \in R} Y_r \geq T_2\right] \quad \leq \quad \left(\mathrm{e}M\left(\frac{N}{n}\right)^{d-1}\right)^{T_2}.$$

We want the probability on the left to be at most, say, $O(\frac{1}{n^c})$ for some constant $c \geq 1$. Hence we require

$$\left[\mathrm{e}M\left(\frac{N}{n}\right)^{d-1}\right]^{T_2} \quad \leq \quad \frac{1}{n^c}.$$

We now take logarithms of both sides and remove lower order terms. Note that as $M = O(\log n)$ its contribution is only a lower order term. Simplifying yields:

$$T_2\left(T_1 \log T_1 - T_1 \log \frac{md}{n}\right) \quad \geq \quad \frac{c \log n}{d-1}. \tag{2.10}$$

For $m = n$, we may choose $T_1 = T_2 = \Theta\left(\sqrt{\frac{\log n}{\log\log n}}\right)$, and the result follows. ∎

One would be inclined to believe that increasing d would decrease the final load. The equation (2.10) indicates that this is true when $m = n$ for very large n, as in this case the effect of d is to reduce the required values of T_1 and T_2 by a constant factor. In practice, however, for reasonable values of n, increasing d yields no improvement, and in fact increasing d can increase the final load. This can be explained by the term $-T_1 \log \frac{md}{n}$ in equation (2.10), which has a disproportionate effect when T_1 and T_2 are small. Also, the constant factor is dictated by our attempt to have the probability of failure be at most $O\left(\frac{1}{n}\right)$; if one is willing to accept slightly larger error probabilities one can improve it slightly.

2.4.2 Multiple round strategies

Our lower bounds suggest that with more rounds of communication, one might achieve a better load distribution. We thus suggest an alternative parallelization of GREEDY called MPGREEDY that makes use of more rounds. Although this algorithm may not be useful in practical settings, its connection to the GREEDY scheme appears interesting in its own right.

The algorithm proceeds in a number of rounds, until every ball has committed. In each round, each bin will allow at most one of its requesting balls to commit to it. If a ball receives that privilege from more than one bin, the ball commits to the bin with the smallest current load. Once a ball has committed, the bins holding the other requests are informed that they may discard those requests:

MPGREEDY(d):
call CHOOSE(d)
in parallel: each ball a
chooses a random I.D. number
sends requests with I.D. to bins $i_1(a), \ldots, i_d(a)$
in parallel: each bin i
sorts requests by I.D. number
sequentially: repeat until all balls have committed
in parallel: each bin i
sends current load to first uncommitted ball on request list
in parallel: each ball a
if received at least one message
commits to the bin with smallest current load
tells bins holding other requests to discard

One can imagine the algorithm by picturing a *scanline* moving level by level up the request lists of the bins. When the scanline moves up to a new level, bins send messages to all the balls that the scanline has just passed through. When bins receive responses, they delete the corresponding balls in the request list above the scanline. The algorithm terminates when every request has either been passed through or deleted.

A practical disadvantage of this algorithm is that it requires synchronous rounds; the discards for each round must complete before the next round can begin. We also require a partial order on the balls, given in this case by randomly chosen I.D. numbers (chosen from a suitably large set to ensure uniqueness with high probability), to instill some notion of sequentiality.

Clearly, the maximum number of balls in any bin upon completion is bounded above by the number of rounds taken to finish. We analyze the latter.

Theorem 2.16 *With high probability* MPGREEDY(d) *finishes in at most* $\frac{\log\log n}{\log d} + 2d + O(1)$ *rounds, and hence the maximum load is also* $\frac{\log\log n}{\log d} + 2d + O(1)$.

In order to prove the above statement, we relate MPGREEDY to the following variant of GREEDY (for any $d \geq 2$): if, when placing a ball, there is a tie for the least loaded bin,

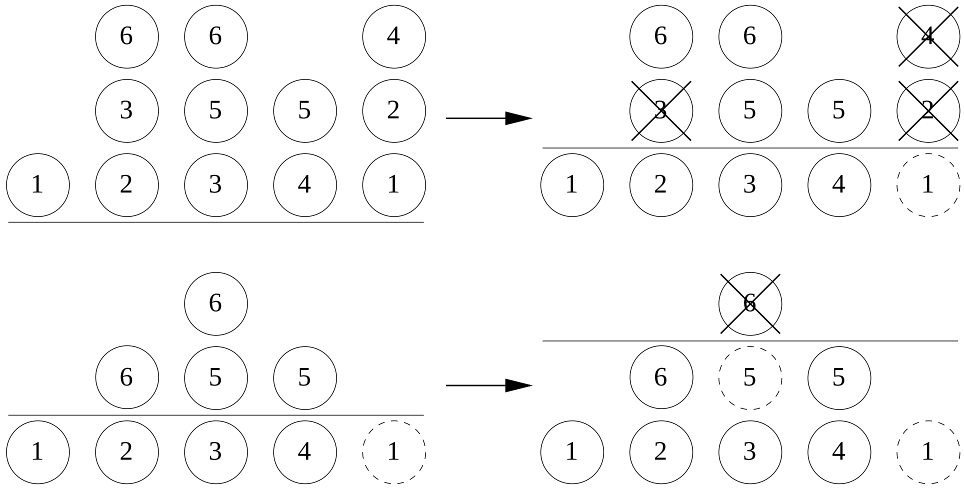

Figure 2.4: Comparing GREEDY WITH TIES and MPGREEDY. Each level is one round of communication. The crossed and dashed balls are discarded by the MPGREEDY process. The GREEDY WITH TIES process includes the dashed balls.

then a copy of the ball is placed in each bin with the minimal load. We call this scheme GREEDY WITH TIES.

Lemma 2.17 *The number of communication rounds used by* MPGREEDY *is one more than the maximum load produced by* GREEDY WITH TIES *when the balls are thrown in the order given by the I.D. numbers and the bin choices made by the balls are the same for both trials.*

Proof: Consider a modification of MPGREEDY where a ball commits to all bins from which it receives a message. The number of communication rounds used by this modified version of MPGREEDY is the same as for the original. With a little thought one can see that this scheme exactly mimics the GREEDY WITH TIES scheme, and hence the two methods give the same final distribution of the balls. (See Figure 2.4.) Since the height of the scanline moves up one level each round, the number of communication rounds used by MPGREEDY is one more than the maximum load of GREEDY WITH TIES. ■

Remark: Using ties as we have done may seem unnecessary, but it allows the scanline to be at the same height for all bins after each round. It may appear that it would improve the final maximum load if, after ties are deleted, heights in the request lists are updated to reflect the deletions. This is difficult to prove, because once heights are updated in this way, the connection between the scanline scheme and the GREEDY scheme is not readily apparent. ■

We now suggest a modification of the proof of Theorem 1.1 to handle the case where there may be ties. The following statement is sufficient:

Theorem 2.18 *The maximum load achieved by* GREEDY WITH TIES *when n balls are thrown into n bins is at most* $\frac{\log\log n}{\log d} + 2d + O(1)$ *with high probability. In particular, for any fixed d the maximum load is* $\frac{\log\log n}{\log d} + O(1)$.

Proof: The proof is almost entirely the same as Theorem 1.1: we review the differences here. Recall that h_t is the height of the tth ball, and $\nu_i(t)$ and $\mu_i(t)$ refer to the number of bins with load at least i and the number of balls with height at least i at time t, respectively. The main consideration is that for each ball placed in the system up to d copies can be placed if ties remain. As before we let $\mathcal{E}_i$ be the event that $\nu_i(n) \leq \beta_i$, but we must use a different inductive definition of β_i, As our base case, we may take $\beta_{6d^2} = n/2d\mathrm{e}$; then $\mathcal{E}_{6d^2}$ holds with certainty. We set $\beta_{i+1} = \mathrm{e}d\beta_i^d/n^{d-1}$.

For a fixed i consider a sequence of random variables Y_t where

$$Y_t = \begin{cases} d & \text{iff } h_t \geq i+1 \text{ and } \nu_i(t-1) \leq \beta_i, \\ 0 & \text{otherwise.} \end{cases}$$

Note that over any given set of choices for the balls before time t, $\mathbf{Pr}(Y_t = d) \leq (\beta_i/n)^d = p_i$; hence by Lemma 1.2

$$\mathbf{Pr}(\textstyle\sum Y_t \geq k) \leq \mathbf{Pr}\left[B(n, p_i) \geq \frac{k}{d}\right],$$

where $B(n,p)$ is the sum of n independent Bernoulli random variables. Conditioned on $\mathcal{E}_i$ we have $\nu_{i+1} \leq \mu_{i+1} \leq \sum Y_t$, so

$$\begin{aligned} \mathbf{Pr}(\nu_{i+1} \geq k \mid \mathcal{E}_i) &\leq \mathbf{Pr}(\textstyle\sum Y_t \geq k \mid \mathcal{E}_i) \\ &\leq \mathbf{Pr}(B(n,p_i) \geq \frac{k}{d} \mid \mathcal{E}_i) \\ &\leq \frac{\mathbf{Pr}(B(n,p_i) \geq k/d)}{\mathbf{Pr}[\mathcal{E}_i]} \end{aligned}$$

The proof now proceeds along the same lines as that of Theorem 1.1. This shows that the maximum load $\frac{\log\log n}{\log d} + 6d^2 + O(1)$. We can improve this by taking a different base case: for $d > 8$, $\beta_{2d} = n/2de$ holds with high probability even if all balls are simply placed into d random bins, and hence we can start the induction from this point instead. ■

Theorem 2.16 follows immediately from Lemma 2.17 and Theorem 2.18. Moreover, an extension to the case where d grows with n is interesting.

Corollary 2.19 *When* MPGREEDY *is run with* $d = \frac{\log\log n}{\log\log\log n} + O(1)$*, the number of rounds and maximum load are at most* $O(\frac{\log\log n}{\log\log\log n})$ *with high probability.* ■

Theorem 2.18 demonstrates that one can match the performance of GREEDY using only $\frac{\log\log n}{\log d} + 2d + O(1)$ rounds of communication, rather than the obvious n rounds. Corollary 2.19 also matches the lower bound of Corollary 2.10, up to constant factors.

It is an open question whether one can extend MPGREEDY to avoid the need for the partial order on the balls or the synchronous rounds while achieving a similar maximum load. Stemann, using a different algorithm, also achieves a maximum load as good as the MPGREEDY algorithm [73]. This algorithm is also not completely asynchronous, although it seems to require weaker synchrony than MPGREEDY.

2.5 The threshold strategy

We now examine another strategy, previously exploited in [24] and [42] in similar contexts, to achieve good load balancing. Given a threshold T, we imagine throwing the balls over r rounds. If more than T balls enter a bin during a round, the excess balls are rethrown in the next round. We wish to set T as small as possible while ensuring that, with high probability, at most T balls are thrown into any bin in the rth round. Then, after all r rounds, the fullest bin will contain at most rT balls. Note that a ball can choose its bins for all r rounds before any messages are sent, so this scheme again falls into the general model of Section 2.2 for which our lower bounds apply.

There are several advantages this method has over the PGREEDY strategy already presented. First, this method can work in completely asynchronous environments: as long as a request includes the number of its current round as part of the message, messages from distinct rounds can be handled simultaneously at the bins. Secondly, balls send and receive at most one message per round. Finally, as we shall show, this method demonstrates a potentially useful tradeoff between the maximum load and the number of rounds.

```
THRESHOLD(T):
  while there exists a ball that has not been accepted
    in parallel: each unaccepted ball a
      chooses u.a.r. a bin i(a)
      sends a request to i(a)
    in parallel: each bin i
      chooses max{T, #received} requests from current round
      sends these balls acceptances
      sends other requesting balls rejections
```

The question is how to set the parameter T so that the procedure terminates with high probability within some specified number of rounds. In Section 2.5.1, we show how to set T for any constant number of rounds. We then show in Section 2.5.2 that, when $T = 1$, THRESHOLD(T) takes at most $O(\log\log n)$ rounds and has maximum load $\Omega(\log\log n)$ with high probability. Our proofs demonstrate the essential techniques needed to derive the relationship between T and r for any values of T and r.

2.5.1 Thresholds with a fixed number of rounds

Theorem 2.20 *If r is fixed independent of n, then* THRESHOLD(T) *terminates after r rounds with high probability, where $T = O\left(\sqrt[r]{\frac{\log n}{\log\log n}}\right)$.*

Proof: Let k_i be the number of balls to be (re)thrown after i rounds ($k_0 = n$). We will show by induction that

$$k_i \leq n\left(\frac{4\log n}{T!}\right)^{\frac{T^i-1}{T-1}} \tag{2.11}$$

for all $i \leq r-1$ with high probability. From this statement one may verify that for constants r and $0 < \epsilon < 1$, and suitably large n, $T = O\left(\sqrt[r]{\frac{\log n}{\log\log n}}\right)$ suffices to reduce k_{r-1} to less than $n^{1-\epsilon}$. We may then conclude that only one more round is necessary by applying Lemma 2.13 with $m = n^{1-\epsilon}$.

We now inductively prove equation (2.11). The case $i = 0$ is readily verified. Now consider the situation when k_i balls are thrown into n bins in the $(i+1)$-st round. It can be verified from equation (2.11) that for large enough n, $k_i/n < T$ for all $i \in \{0, \ldots, r\}$. We can thus apply the Poisson approximation and Corollary 2.12 to obtain that, with high probability, in the $(i+1)$-st round,

$$\mathbf{Pr}(\text{a given bin receives} > T \text{ requests}) \quad \leq \quad \frac{2e^{-k_i/n}(k_i/n)^T}{T!}. \tag{2.12}$$

Therefore (via the Chernoff bounds of Lemma 1.3) with high probability the number of bins with more than T requests is at most $\frac{4ne^{-k_i/n}(k_i/n)^T}{T!}$. We can make the conservative upper bound assumption that with probability exponentially close to one, none of these over-full bins has more than $\log n$ requests, so that with high probability,

$$k_{i+1} \quad \leq \quad n \left(\frac{4 \log n}{T!} \right)^{\frac{T(T^i-1)}{T-1}+1}. \tag{2.13}$$

Equation (2.11) now follows by induction, as long as the number of rethrows is large enough so that the Chernoff bound holds. This immediately implies a maximum load of $O(rT)$, which, for fixed r, is $O\left(\sqrt[r]{\frac{\log n}{\log \log n}}\right)$. ■

The theorem suggests that using the threshold strategy, one can successfully trade load balance for communication time in a well-defined manner. We note that one can also show that for $T = \Omega\left(\sqrt[r]{\frac{\log n}{\log \log n}}\right)$, THRESHOLD($T$) requires more than r rounds with high probability. We also remark that we could show that Theorem 2.20 holds with very high probability; that is, the probability of failure is bounded above by $1/f(n)$ where $f(n)$ is a superpolynomial function. This requires more attention to the Chernoff bounds.

2.5.2 The case of $T = 1$

We can extend our argument to the case where r grows with n with a bit more care. As an illustrative example, we consider the case where $T = 1$. We note that similar more powerful results are given in [42] and [55], but the simple proofs below are appealing.

Theorem 2.21 THRESHOLD(1) *terminates after at most* $\log \log n + O(1)$ *stages with high probability.*

Proof: As in the proof of Theorem 2.20, let k_i be the number of balls to be thrown after round i. It is simple to show by Chernoff bounds (equation (1.1)) that, with high probability, after only two rounds at most $n/2\mathrm{e}$ balls remain to be thrown. We claim that, as long as k_{i+1} is at least $4\sqrt{n \log n}$, $k_{i+1} \le \mathrm{e}k_i^2/n$ with probability $1-O(1/n^2)$. For convenience we assume that in each round the balls arrive in some arbitrary order, with the first ball that arrives at a bin being accepted. Let X_j be the indicator random variable of the event that the jth ball falls into a non-empty bin, where $1 \le j \le k_i$. Note that $\mathbf{Pr}(X_j = 1 \mid X_1, \ldots, X_{j-1}) \le k_i/n$. It follows from Lemma 1.2 that the sum of the k_i random variables X_j is stochastically dominated by the sum of k_i independent Bernoulli random variables with parameter k_i/n. Using Chernoff bounds (equation (1.1)) the above claim follows; the restriction that k_{i+1} is at least $4\sqrt{n \log n}$ is required for the Chernoff bound to hold with sufficient probability. We thus have, if $i \ge 2$ and $k_i \ge 4\sqrt{n \log n}$, that

$$k_i \le \frac{n}{\mathrm{e}2^{2^{i-2}}}.$$

Hence $r = \log\log n + O(1)$ rounds will suffice to cut down k_r to below $4\sqrt{n \log n}$ with high probability. By using the Poisson case to bound the number of bins that receive more than one ball, one can show that only $O(1)$ more rounds will be needed after this point, and the result follows. ∎

The strategy THRESHOLD(1) achieves a maximum load that is essentially the same as GREEDY, but uses only $O(\log\log n)$ asynchronous rounds instead of $O(n)$ synchronous rounds. Moreover, because of its simplicity, we expect that this strategy may be the best choice when the GREEDY strategy does not apply. One might hope that the bound of Theorem 2.21 is not tight, and that THRESHOLD(1) actually matches the lower bound of Corollary 2.10. This could happen in one of two ways: either THRESHOLD(1) might terminate in fewer than $\Omega(\log\log n)$ rounds, or even if $\Omega(\log\log n)$ rounds are required, perhaps no bin actually receives $\Omega(\log\log n)$ balls. We will now show, however, that the bound of Theorem 2.21 is tight, up to constant factors.

Theorem 2.22 *The maximum load of* THRESHOLD(1) *is at least* $\Omega(\log\log n)$ *with high probability.*

Proof: As before, let k_i be the number of balls to be thrown in round i, with $k_0 = n$. We can determine k_{i+1} by considering the number of bins that receive two or more balls in the ith round. In the Poisson case, the probability that a bin receives two balls in round

i is $e^{-k_i/n}\frac{k_i^2}{2n^2} \geq \frac{k_i^2}{2en^2}$. By equation (1.2) of Lemma 1.3 and Corollary 2.12, as long as $k_i > 10\sqrt{n \log n}$, then with probability at least $1 - O(1/n^2)$, $k_{i+1} \geq \frac{k_i^2}{4en}$. Hence, for all $i \leq n$ with $k_i > 10\sqrt{n \log n}$,

$$k_{i+1} \geq \left(\frac{1}{4en}\right)^{2^{i+1}-1} k_0^{2^{i+1}} = \frac{4en}{(4e)^{2^{i+1}}}. \tag{2.14}$$

It is easy to check from equation (2.14) that we need $i = \Omega(\log\log n)$ before $k_i \leq 10\sqrt{n \log n}$. We now show that with high probability, there will be at least one bin that receives a ball in each of the first $\Omega(\log\log n)$ rounds. Say that a bin *survives* up to round i if it gets a ball in each of rounds $1, \ldots, i$, and let s_i be the number of bins that survive up to round i. Then

$$\mathbf{Pr}\Big[\text{bin survives up to } i+1 \,\Big|\, \text{it survives up to } i\Big] = 1 - \left(1 - \frac{1}{n}\right)^{k_i} \geq \frac{k_i}{2n},$$

where the last inequality holds since $k_i \leq n$. Applying Chernoff's bound (equation (1.2)) tells us that the fraction of bins that survived round i that also survive round $i+1$ is at least $\frac{k_i}{4n}$ with probability at least $1 - O(\frac{1}{n^2})$ as long as s_i is sufficiently large. Therefore, after the $(i+1)$-st round, with high probability the number of surviving bins is at least

$$\begin{aligned} s_{i+1} &\geq n \times \frac{k_0}{4n} \times \cdots \times \frac{k_i}{4n} \\ &> \frac{ne^{i+1}}{(4e)^{2^{i+1}}}. \end{aligned}$$

It remains to be checked that for $i = \Omega(\log\log n)$ all the Chernoff bounds will hold, and thus with high probability there is still a surviving bin. ■

2.6 Simulation results

It is important to note that in the balls and bins scenario, even if each ball just chooses one bin independently and uniformly at random, the maximum load is very small compared to the total number of bins. Thus, even though one may be able to show that asymptotically one strategy performs better than another, it is worthwhile to test actual performance. For example, it is not clear from the results we have described that GREEDY performs better than straightforward random selection unless n is exceedingly large! (In fact, for all values of n, the expected maximum load of GREEDY is less than that of simple random selection; see [11] for more details.) Even if one can guarantee better performance, however, a system designer interested in using a load balancing scheme must balance the tradeoff between the maximum load and the complexity of the underlying algorithm. Asymptotic notation proves less helpful than specific numbers in understanding this tradeoff. We therefore examine actual performance through some simulation results.

For simplicity, we here consider only the case where the numbers of balls and bins are equal. As usual, d represents the number of bins to which each ball sends requests. The numbers given in the table represent the maximum load found for one hundred trials of each strategy.

The first thing worth noting is that GREEDY performs noticeably better than simple one-choice randomization: even at just one million balls, the difference is at least a factor of two in all of our trials. A second interesting feature of GREEDY is that the maximum load appears to vary little across trials, suggesting that the maximum load is sharply concentrated on a single value. We shall gain more insight into this phenomenon in Section 4.2.

As expected, both PGREEDY and THRESHOLD(T) perform somewhere between simple random selection and GREEDY. Notice that for PGREEDY when $d = 3$ the maximum load tends to be smaller than when $d = 2$, but that the maximum load tends to increase when $d = 4$. This is not completely surprising given our previous analysis in Section 2.4.

In the threshold schemes, the thresholds used were as follows: 3 balls per round per bin in the 2 round scheme, and 2 balls per bin per round in the 3 round scheme. These choices were made to ensure that the algorithm terminated with all balls having a final destination in the correct number of rounds with high probability: in all trials, the algorithm terminated in the correct number of rounds. Our simulations suggest that threshold schemes are the best practical choice when one wishes to achieve a better load balance, but cannot meet the sequentiality requirement of GREEDY.

Balls	One	GREEDY			PGREEDY			THRESHOLD(T)	
n	Choice	$d=2$	$d=3$	$d=4$	$d=2$	$d=3$	$d=4$	2 rnds.	3 rnds.
1 m.	**8** ..28 **9** ..57 **10**..13 **11**..2	**4**..100	**3**..100	**3**..100	**5**..92 **6**..8	**5**..95 **6**..5	**5**..77 **6**..23	**5**..88 **6**..12	**4**..77 **5**..23
2 m.	**8** ..7 **9** ..72 **10**..18 **11**..3	**4**..100	**3**..100	**3**..100	**5**..90 **6**..10	**5**..96 **6**..4	**5**..68 **6**..32	**5**..74 **6**..26	**4**..69 **5**..31
4 m.	**8** ..1 **9** ..63 **10**..35 **12**..1	**4**..100	**3**..100	**3**..100	**5**..71 **6**..29	**5**..87 **6**..13	**5**..36 **6**..64	**5**..54 **6**..46	**4**..47 **5**..53
8 m.	**9** ..40 **10**..58 **11**..1 **12**..1	**4**..100	**3**..100	**3**..100	**5**..55 **6**..45	**5**..71 **6**..29	**5**..6 **6**..94	**5**..20 **6**..80	**4**..19 **5**..81
16 m.	**9** ..21 **10**..62 **11**..16 **16**..1	**4**..100	**3**..100	**3**..100	**5**..31 **6**..69	**5**..48 **6**..52	**5**..1 **6**..99	**5**..5 **6**..95	**4**..1 **5**..99
32 m.	**9** ..5 **10**..65 **11**..25 **12**..5	**4**..100	**3**..100	**3**..100	**5**..8 **6**..92	**5**..20 **6**..80	**6**..100	**6**..100	**5**..100

Table 2.1: Simulation results for GREEDY and other strategies. The number of balls ranges from one million to thirty-two million. The results from 100 trials are presented: the load is given in bold on the left, and the frequency of that load is given on the right.

Chapter 3

The supermarket model

3.1 Introduction

3.1.1 The problem and model

In this chapter, we move from considering static problems to dynamic problems. Recall that in dynamic models, tasks arrive and leave over time. Dynamic models often capture more realistic settings than static models. For example, in the task-processor model described in Section 1.1.2, there may not be a fixed number of tasks to distribute, but instead tasks may arrive at the system over time and run for a certain period before completing. Similarly, in the hashing model described Section 1.1.2, the hash table may not have a fixed number of entries, and items may be deleted as well as placed in the hash table over time. Our goal in studying these systems will be to determine properties of the system in equilibrium, or over arbitrarily long periods of time.

In looking for a dynamic generalization of balls and bins problems, we are naturally led to examine queueing models. We will assume that the reader has some familiarity with the basic terminology and results from queueing theory, which can be found in most standard introductory texts on stochastic processes (*e.g.* [66, 67, 76]). In particular, we expect a basic understanding of the M/M/1 queue.

We first develop the appropriate techniques by focusing on the following natural dynamic model: customers arrive as a Poisson stream of rate λn, where $\lambda < 1$, at a collection of n servers. Each customer chooses some constant number d of servers independently and uniformly at random from the n servers, and waits for service at the server currently

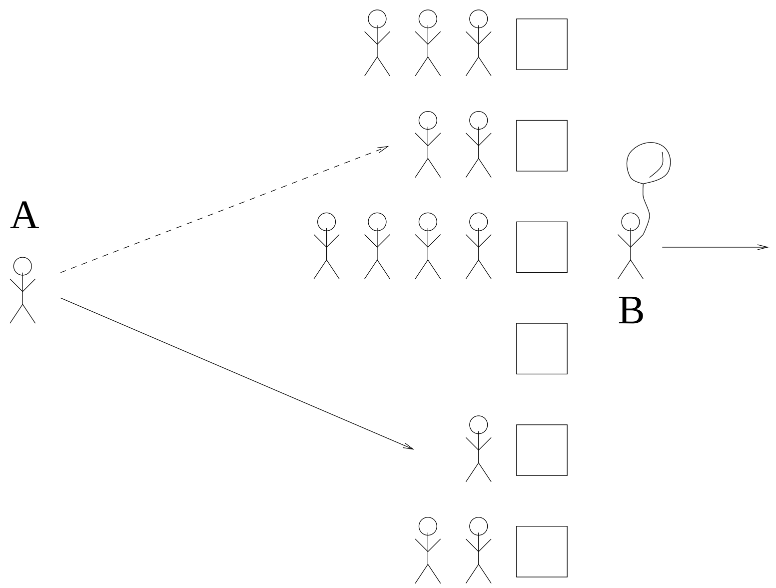

Figure 3.1: The supermarket model. Incoming customer A chooses two random servers, and queues at the shorter one. Customer B has recently been served and leaves the system.

containing the fewest customers (ties being broken arbitrarily). Customers are served according to the First In First Out (FIFO) protocol, and the service time for a customer is exponentially distributed with mean 1. We call this model the *supermarket model*, or the *supermarket system* (see Figure 3.1). We are interested in the expected time a customer spends in the system in equilibrium, which we claim is a natural measure of system performance. Note that the average arrival rate per queue is $\lambda < 1$, and that the average service rate is 1; hence we expect the system to be *stable*, in the sense that the expected number of customers per queue remains finite in equilibrium.

In this simple model, we have assumed that the time for a customer to obtain information about queue lengths at the servers and the time to move to a server is zero. Also, we have assumed that the processors are *homogeneous*, in that the service rates are the same at each server. These assumptions are made for clarity and ease of presentation; most of our techniques generalize easily to more complex systems.

As in the static case, the supermarket model is much easier to analyze when $d = 1$. In this case, the arrival stream can be split into independent Poisson streams for each server, and hence each server acts as a simple M/M/1 queue. For $d \geq 2$, the supermarket model proves difficult to analyze because of dependencies: knowing the length of one queue affects the distribution of the length of all the other queues.

As we have described it, the supermarket model is a *Markov process*. That is, the future of the supermarket system depends only on its current state, and not on the past. The Markovian nature of the supermarket model will allow us to apply sophisticated techniques from the theory of Markov chains to the problem. The fact that the supermarket model is expressible as a suitable Markov process depends on our assumptions of Poisson arrivals and exponential service times, but in practice, such strong assumptions are often unfounded. This issue will be covered in the next chapter, where we consider how to approximate non-Markovian models with Markovian models, allowing us to apply these techniques in the study of non-Markovian systems.

Although it appears to be a natural model, there has been little reference to the supermarket model in previous literature. As mentioned in Chapter 1, a great deal of work has been done to study the model where incoming customers join the shortest queue, the most recent of which is due to Adan and others [3, 4, 5]. The limited coordination enforced by our model corresponds nicely to models of *distributed* systems, as distinguished from *centralized* systems, where the shortest queue model appears more applicable. The supermarket model has been studied both analytically and with simulations by Eager *et al.* [29] and through trace-driven simulations by Zhou [77]. Both works demonstrate the effectiveness of each customer having a small number of choices. The analytic results of Eager *et al.*, however, are derived using the assumption that the state of each queue in the supermarket model in stochastically independent of the state of any other queue [29, p. 665]. The authors assert (without justification) that this approach is exact in the asymptotic limit as the number of queues grows to infinity. In contrast, we do not begin with assumptions of independence, and one of our main results verifies the assertion of Eager *et al.* regarding the asymptotic limit.

3.1.2 Methodology and results

Our results will be based on the following approach:

- We define an idealized process, corresponding to a system of infinite size. We then analyze this process, which is cleaner and easier because its behavior is completely deterministic.

- We relate the idealized system to the finite system, carrying over the analysis with bounded error.

For example, in the supermarket model, the intuition is that if we look at the *fraction* of servers containing at least k customers for every k, the system evolves in an almost deterministic way as $n \to \infty$. The analysis of this system is interesting in its own right. Then we bound the deviation between a system of finite size n and the infinite system.

The following result is typical of our method:

Theorem: *For any fixed T and $d \geq 2$, the expected time a customer spends in the supermarket system when it is initially empty over the first T units of time is bounded above by*

$$\sum_{i=1}^{\infty} \lambda^{\frac{d^i - d}{d-1}} + o(1),$$

where the $o(1)$ term is understood as $n \to \infty$ (and may depend on T).

The summation is derived by studying the infinite system, and the $o(1)$ error term arises when we bound the error between the infinite system and the system for a fixed n. The combination of the two analyses yields the theorem. This result should be compared to the case where $d = 1$, where the expected time is $1/(1-\lambda)$ in equilibrium. As we shall see in Section 3.3.4, for λ close to 1 there is an exponential improvement in the expected time a customer spends in the system when $d \geq 2$.

Besides providing the first analysis of the supermarket model, we note that this approach also provides a clean, systematic approach to analyzing several other load balancing models, as we will see in the next chapter. Further, the method provides a means of finding accurate numerical estimates of performance. In Section 3.6 we present simulation results to demonstrate the accuracy of our approach.

To bound the error between the finite and infinite systems we will use Kurtz's work on *density dependent jump Markov processes* [32, 49, 50, 51, 52], with some extensions specific to our problems. Kurtz's work has previously been applied to matching problems on random graphs [38, 43, 44] as well as some queueing models [70]; here, we apply it for the first time to load balancing problems. Given the increasing use of Markov chains in the analysis of algorithms, we believe that this technique may be more widely applicable than previously expected.

The rest of this chapter is structured as follows: we first present an intuitive example of the infinite system approach, based on an epidemic model. Then, in Section 3.3, we derive and analyze the behavior of the infinite version of the supermarket model. In Section 3.4, we explain Kurtz's work and how to adapt it to relate the finite and infinite versions of the supermarket model, as well as offer a generalization of his main theorem to certain infinite dimensional problems. In Section 3.5 we apply Kurtz's theorem to the supermarket model to bound the expected time a customer spends in the system. Finally, in Section 3.6, we provide simulation results demonstrating the effectiveness of the infinite system approach.

3.2 Infinite systems: the epidemic example

To explain the infinite system approach, we begin with a simple example due to Kurtz [52] using a model of the behavior of epidemics familiar to students of second year calculus. We assume that the rate at which susceptible people become infected is proportional to the amount of interaction between the susceptible and infected population, and that infected people recover and become immune independently at a fixed rate.

A sophisticated attack would model the problem as a Markov chain. We introduce a parameter N, corresponding to the population size. We take as our state space pairs (X, Y), where X is the number of susceptible people, Y is the number of infected people, and $N - X - Y$ is the number of immune people. The transition intensities q are given by the equations

$$\begin{aligned} q_{(X,Y),(X-1,Y+1)} &= -\lambda X \frac{Y}{N} = -N\lambda \frac{X}{N}\frac{Y}{N}; \\ q_{(X,Y),(X,Y-1)} &= -\mu Y = -N\mu \frac{Y}{N}. \end{aligned}$$

Here λ and μ are fixed constants.

A second year calculus student would instead model the problem (without formal justification) by the deterministic solution to a set of differential equations. Let x be the fraction of the population that is susceptible to the disease, and y be the fraction of the population that is infected. The following differential equations intuitively correspond to the description of the model:

$$\frac{dx}{dt} = -\lambda xy; \tag{3.1}$$

$$\frac{dy}{dt} = \lambda xy - \mu y. \tag{3.2}$$

Noting that $x = \frac{X}{N}$ and $y = \frac{Y}{N}$, we can see the relationship between the deterministic process given by the differential equations and the random process given by the Markov chain. Namely, for a small time interval Δt, we have

$$\mathsf{E}[\Delta X] = -N\lambda \frac{X}{N}\frac{Y}{N}\Delta t\,; \tag{3.3}$$

$$\mathsf{E}[\Delta Y] = \left(N\lambda \frac{X}{N}\frac{Y}{N} - N\mu\frac{Y}{N}\right)\Delta t. \tag{3.4}$$

If we modify the state of the Markov chain to record the fractions (x, y) instead of the numbers (X, Y), then the differential equations (3.1) and (3.2) describe the expected behavior of the Markov chain over a small interval of time as described in equations (3.3) and (3.4). Hence, one might expect that the path taken by the Markov chain should look something like the path given by the differential equations.

In fact the deterministic path given by the differential equations describes almost surely the limiting behavior of the Markov chain as N, the population size, grows to infinity. This should not be surprising: as N grows to infinity, a law of large numbers for Markov processes takes effect, and the system must behave close to its expectation. This informal intuition that looking at differential equations describes the expected behavior of the system will be justified later by Kurtz's theorem in Section 3.4.1.

3.3 The analysis of the supermarket model

3.3.1 Preliminaries

Recall the definition of the supermarket model: customers arrive as a Poisson stream of rate λn, where $\lambda < 1$, at a collection of n FIFO servers. Each customer chooses some constant $d \geq 2$ servers independently and uniformly at random with replacement[1] and queues at the server currently containing the fewest customers. The service time for a customer is exponentially distributed with mean 1. The following intuitive lemma, which we state without proof, will be useful:

Lemma 3.1 *The supermarket system is stable for every $\lambda < 1$; that is, the expected number of customers in the system remains finite for all time.* ■

Remark: Lemma 3.1 can be proven by a simple comparison argument with the system in which each customer queues at a random server (that is, where $d = 1$); in this system,

[1] We note that our results also hold with minor variations if the d queues are chosen without replacement.

each server acts like an M/M/1 server with Poisson arrival rate λ, which is known to be stable (see, for example, [46]). The comparison argument is entirely similar to those in [74] and [75], which show that choosing the shortest queue is optimal subject to certain assumptions on the service process; alternatively, an argument based on majorization is given in [10]. We also remark that a similar argument shows that the size of the longest queue in a supermarket system is stochastically dominated by the size of the longest queue in a set of n independent M/M/1 servers. ∎

We now introduce a representation of the system that will be convenient throughout our analysis. We define $n_i(t)$ to be the number of queues with i customers at time t; $m_i(t)$ to be the number of queues with at least i customers at time t; $p_i(t) = n_i(t)/n$ to be the fraction of queues of size i; and $s_i(t) = \sum_{k=i}^{\infty} p_i(t) = m_i(t)/n$ to be the tails of the $p_i(t)$. We drop the reference to t in the notation where the meaning is clear. As we shall see, the s_i prove much more convenient to work with than the p_i. Note that $s_0 = 1$ always, and that the s_i are non-increasing. In an *empty system*, which corresponds to one with no customers, $s_0 = 1$ and $s_i = 0$ for $i \geq 1$. By comparing this system with a system of M/M/1 queues as in the remark after Lemma 3.1, we have that if $s_i(0) = 0$ for some i, then for all $t \geq 0$ we have that $\lim_{i \to \infty} s_i(t) = 0$. Under the same conditions, the expected number of customers per queue, or $\sum_{i=1}^{\infty} s_i(t)$, is finite even as $t \to \infty$.

We can represent the state of the system at any given time by an infinite dimensional vector $\vec{s} = (s_0, s_1, s_2, \ldots)$. The state only includes information regarding the number of queues of each size; this is all the information we require. It is clear that for each value of n, the supermarket model can be considered as a Markov chain on the above state space. That is, the future state of the system depends only on the current state, and not on the past. Notice that, with this state description, the assumptions of Poisson arrivals and exponential service times ensure that the system is Markovian. If instead the service times were constant, then the time for the next departure would depend on the past, specifically on when all the customers being served began service.

We now introduce a deterministic *infinite system* related to the finite supermarket system. The time evolution of the infinite system is specified by the following set of differential equations:

$$\begin{cases} \dfrac{ds_i}{dt} & = \lambda(s_{i-1}^d - s_i^d) - (s_i - s_{i+1}) \quad \text{for} \quad i \geq 1\,; \\ s_0 & = 1. \end{cases} \tag{3.5}$$

Let us explain the reasoning behind the system (3.5). Consider a supermarket system with n queues, and determine the expected change in the number of servers with at least i customers over a small period of time of length dt. The probability a customer arrives during this period is $\lambda n\,dt$, and the probability an arriving customer joins a queue of size $i-1$ is $s_{i-1}^d - s_i^d$. (This is the probability that all d servers chosen by the new customer are of size at least $i-1$ but not all are of size at least i.) Thus the expected change in m_i due to arrivals is exactly $\lambda n(s_{i-1}^d - s_i^d)dt$. Similarly, the probability a customer leaves a server of size i in this period is $n_i\,dt = n(s_i - s_{i+1})dt$. Hence, if the system behaved according to these expectations, we would have

$$\frac{dm_i}{dt} = \lambda n(s_{i-1}^d - s_i^d) - n(s_i - s_{i+1}).$$

Removing a factor of n from the equations yields the system (3.5). That this infinite set of differential equations has a unique solution given appropriate initial conditions is not immediately obvious; however, it follows from standard results in analysis (see [1, p. 188, Theorem 4.1.5], or [26, Theorem 3.2]). It should be intuitively clear that as $n \to \infty$ the behavior of the supermarket system approaches that of this deterministic system. A formal justification of this relationship will be given in Section 3.4. For now, we simply take this set of differential equations to be the appropriate limiting process.

It is worth noting here that the state space of our infinite system is, in this case, infinite dimensional, since we record s_i for every i. If the queue lengths were bounded, then the system would only be finite dimensional, as we would only need to consider values $s_1, \ldots, s_B$ for some bound B. (Indeed, we will say more about this model in the next chapter.) We point this out because, in some cases, the mathematics of infinite dimensional systems can be much more difficult to deal with.[2] We try not to focus excessively on these technical details, although we will take special care and cite the relevant material where appropriate.

[2] As an example, consider the following simple paradox: place a light at every positive integer on the number line. The lights are hooked up so that the light at i cannot go on until the light at $i+1$ goes on. At time $t = 0$ all lights are off. It would seem that we could conclude that all lights remain off for all time, but the following schedule has all lights on by time $t = 1$: at time $1/i$, turn on light i. Of course if the number of lights is finite the lights must stay off for all time.

3.3.2 Finding a fixed point

In this section we demonstrate that, given a reasonable condition on the initial point $\vec{s}(0)$, the infinite process converges to a *fixed point*. A fixed point (also called an *equilibrium point* or a *critical point*) is a point $\vec{p}$ such that if $\vec{s}(t) = \vec{p}$ then $\vec{s}(t') = \vec{p}$ for all $t' \geq t$. It is clear that for the supermarket model a necessary and sufficient condition for $\vec{s}$ to be a fixed point is that for all i, $\frac{ds_i}{dt} = 0$.

Lemma 3.2 *The system (3.5) with $d \geq 2$ has a unique fixed point with $\sum_{i=1}^{\infty} s_i < \infty$ given by*

$$s_i = \lambda^{\frac{d^i - 1}{d-1}}.$$

Proof: It is easy to check that the proposed fixed point satisfies $\frac{ds_i}{dt} = 0$ for all $i \geq 1$. Conversely, from the assumption $\frac{ds_i}{dt} = 0$ for all i we can derive that $s_1 = \lambda$ by summing the equations (3.5) over all $i \geq 1$. (Note that we use $\sum_{i=1}^{\infty} s_i < \infty$ here to ensure that the sum converges absolutely. That $s_1 = \lambda$ at the fixed point also follows intuitively from the fact that at the fixed point, the rate at which customers enter and leave the system must be equal.) The result then follows from (3.5) by induction. ■

The condition $\sum_{i=1}^{\infty} s_i < \infty$, which corresponds to the average number of customers per queue being finite, is necessary; $(1, 1, \ldots)$ is also a fixed point, which corresponds the number of customers at each queue going to infinity. Given a suitable initial point, however, we know that $\sum_{i=1}^{\infty} s_i(t) < \infty$ for all $t \geq 0$ by Lemma 3.1.

Definition 3.3 *A sequence $(x_i)_{i=0}^{\infty}$ is said to* decrease doubly exponentially *if and only if there exist positive constants $N, \alpha < 1, \beta > 1$, and γ such that for $i \geq N$, $x_i \leq \gamma \alpha^{\beta^i}$.*

It is worth comparing the result of Lemma 3.2 to the case where $d = 1$ (*i.e.*, all servers are M/M/1 queues), for which the fixed point is given by $s_i = \lambda^i$. The key feature of the supermarket system is that for $d \geq 2$ the tails s_i decrease doubly exponentially, while for $d = 1$ the tails decrease only geometrically (or singly exponentially).

3.3.3 Convergence to the fixed point

We now show that every trajectory of the infinite supermarket system converges exponentially to the fixed point of Lemma 3.2 in an appropriate metric. Denote the above fixed point by $\vec{\pi} = (\pi_i)$, where $\pi_i = \lambda^{\frac{d^i-1}{d-1}}$. We shall assume that $d \geq 2$ in what follows unless otherwise specified.

To show convergence, we find a *potential function* (also called a *Lyapunov function* in the dynamical systems literature) $\Phi(t)$ with the following properties:

1. The potential function is related to the distance between the current point on the trajectory and the fixed point.

2. The potential function is strictly decreasing, except at the fixed point.

The intuition is that the potential function shows that the system heads toward the fixed point. By finding a suitable potential function we will also be able to say how fast the system approaches the fixed point. A natural potential function to consider is $D(t) = \sum_{i=1}^{\infty} |s_i(t) - \pi_i|$, which measures the L_1-distance between the two points. Our potential function will actually be a weighted variant of this, namely $\Phi(t) = \sum_{i=1}^{\infty} w_i |s_i(t) - \pi_i|$ for suitably chosen weights w_i.

We begin with a result that shows the system has an invariant, which restricts in some sense how far any s_i can be from the corresponding value π_i.

Theorem 3.4 *Suppose that there exists some j such that $s_j(0) = 0$. Then the sequence $(s_i(t))_{i=0}^{\infty}$ decreases doubly exponentially for all $t \geq 0$, where the associated constants are independent of t. In particular, if the system begins empty, then $s_i(t) \leq \pi_i$ for all $t \geq 0$.*

Note that the hypothesis of Theorem 3.4 holds for any initial state $\vec{s}$ derived from the initial state of a finite system.

Proof: Let $M(t) = \sup_i [s_i(t)/\pi_i]^{1/d^i}$. We first show that $M(t) \leq M(0)$ for all $t \geq 0$. We will then use this fact to show that the s_i decrease doubly exponentially.

A natural, intuitive proof proceeds as follows: in the case where there are a finite number of queues, an inductive coupling argument can be used to prove that if we increase some $s_i(0)$, thereby increasing the number of customers in the system, the expected value of all s_j after any time t increases as well. Extending this to the limiting case as the number of queues $n \to \infty$ (so that the s_j behave according to their expectations), we have that increasing $s_i(0)$ can only increase all the $s_j(t)$ and hence $M(t)$ for all t.

So, to begin, let us increase all the $s_i(0)$ (including $s_0(0)$!) so that $s_i(0) = M(0)^{d^i} \pi_i$. But then it is easy to check that the initial point is a fixed point (albeit possibly with

$s_0 > 1$), and hence $M(t) = M(0)$ in the raised system. We conclude that in the original system $M(t) \leq M(0)$ for all $t \geq 0$.

A more formal proof that increasing $s_i(0)$ only increases all $s_j(t)$ relies on the fact that the ds_i/dt are *quasimonotone*: that is, ds_i/dt is non-decreasing in s_j for $j \neq i$. The result then follows from [26, pp. 70-74].

We now show that the s_i decrease doubly exponentially (in the infinite model). Let j be the smallest value such that $s_j(0) = 0$, which exists by the hypothesis of the theorem. Then $M(0) \leq [1/\pi_{j-1}]^{1/d^{j-1}} < 1/\lambda^{1/(d-1)}$. Since $M(t) \leq M(0)$, $M(0)^{d^i} \geq s_i(t)/\pi_i$ for $t \geq 0$, or

$$s_i(t) \leq \pi_i M(0)^{d^i} = \lambda^{-1/(d-1)}(\lambda^{1/(d-1)}M(0))^{d^i}.$$

Note that $\lambda^{1/(d-1)}M(0) < 1$, since $M(0) < 1/\lambda^{1/(d-1)}$ Hence the s_i decrease doubly exponentially, with $\alpha = \lambda^{1/(d-1)}M(0)$ and $\beta = d$. In particular, if the system begins empty, then $s_i(t) \leq \pi_i$ for all t and i. ■

We now show that the system not only converges to its fixed point, but that it does so *exponentially.*

Definition 3.5 *The potential function Φ is said to* converge exponentially to 0*, or simply to converge exponentially, if $\Phi(0) < \infty$ and $\Phi(t) \leq c_0 e^{-\delta t}$ for some constant $\delta > 0$ and a constant c_0 which may depend on the state at $t = 0$.*

By finding a potential function Φ that converges exponentially to 0 and measures the distance between the current point on the trajectory and the fixed point, we now show that the system converges exponentially quickly to its fixed point.

Theorem 3.6 *Let $\Phi(t) = \sum_{i=1}^{\infty} w_i|s_i(t) - \pi_i|$, where for $i \geq 1$, $w_i \geq 1$ are appropriately chosen constants to be determined. If $\Phi(0) < \infty$, then Φ converges exponentially to 0. In particular, if there exists a j such that $s_j(0) = 0$, then Φ converges exponentially to 0.*[3]

Proof: We shall prove the theorem by showing there exists a constant δ (that will depend only on λ) such that $d\Phi/dt \leq -\delta\Phi$. This suffices to prove the theorem.

[3]For completeness, we note that in earlier versions of this work we made use of a different potential function: $\Psi(t) = \sum_{i=1}^{\infty} \frac{s_i(t)\log(s_i(t)/\pi_i) - s_i(t) + \pi_i}{d^i}$. The interested reader may enjoy showing that $d\Psi/dt \leq 0$, with equality only at the fixed point. We did not find a way to use this potential function to prove exponential convergence, however.

Define $\epsilon_i(t) = s_i(t) - \pi_i$. As usual, we drop the explicit dependence on t when the meaning is clear. For convenience, we assume that $d = 2$; the proof is easily modified for general d.

As $d\epsilon_i/dt = ds_i/dt$, we have from (3.5)

$$\begin{aligned} \frac{d\epsilon_i}{dt} &= \lambda[(\pi_{i-1} + \epsilon_{i-1})^2 - (\pi_i + \epsilon_i)^2] - (\pi_i + \epsilon_i - \pi_{i+1} - \epsilon_{i+1}) \\ &= \lambda(2\pi_{i-1}\epsilon_{i-1} + \epsilon_{i-1}^2 - 2\pi_i\epsilon_i - \epsilon_i^2) - (\epsilon_i - \epsilon_{i+1}), \end{aligned}$$

where the last equality follows from the fact that $\vec{\pi}$ is the fixed point.

As $\Phi(t) = \sum_{i=1}^{\infty} w_i|\epsilon_i(t)|$, the derivative of Φ with respect to t, $d\Phi/dt$, is not well defined if $\epsilon_i(t) = 0$. We shall explain how to cope with this problem at the end of the proof, and we suggest the reader proceed by temporarily assuming $\epsilon_i(t) \neq 0$.

Now

$$\begin{aligned} \frac{d\Phi}{dt} &= \sum_{i:\epsilon_i>0} w_i[\lambda(2\pi_{i-1}\epsilon_{i-1} + \epsilon_{i-1}^2 - 2\pi_i\epsilon_i - \epsilon_i^2) - (\epsilon_i - \epsilon_{i+1})] - \\ &\quad \sum_{i:\epsilon_i<0} w_i[\lambda(2\pi_{i-1}\epsilon_{i-1} + \epsilon_{i-1}^2 - 2\pi_i\epsilon_i - \epsilon_i^2) - (\epsilon_i - \epsilon_{i+1})]. \end{aligned} \tag{3.6}$$

Let us look at the terms involving ϵ_i in this summation. (Note: ϵ_1 terms are a special case, which can be included in the following if we take $w_0 = 0$. This has no effect on the value of Φ.) There are several cases, depending on whether $\epsilon_{i-1}, \epsilon_i$, and ϵ_{i+1} are positive or negative. Let us consider the case where they are all negative (which, by Theorem 3.4, is always the case when the is initially empty). Then, by equation (3.6), the term involving ϵ_i in $d\Phi/dt$ is

$$-w_{i-1}\epsilon_i + w_i(2\lambda\pi_i\epsilon_i + \lambda\epsilon_i^2 + \epsilon_i) - w_{i+1}(2\lambda\pi_i\epsilon_i + \lambda\epsilon_i^2). \tag{3.7}$$

We wish to choose w_{i-1}, w_i, and w_{i+1} so that this term is at most $\delta w_i\epsilon_i$ for some constant $\delta > 0$. It is sufficient to choose them so that

$$(w_i - w_{i-1}) + (2\lambda\pi_i + \lambda\epsilon_i)(w_i - w_{i+1}) \geq \delta w_i\,;$$

or, using the fact that $|\epsilon_i| \leq 1$,

$$w_{i+1} \leq w_i + \frac{w_i(1-\delta) - w_{i-1}}{\lambda(2\pi_i + 1)}.$$

We note that the same inequality would be sufficient in the other cases as well: for example, if all of $\epsilon_{i-1}, \epsilon_i$, and ϵ_{i+1} are positive, the above term (3.7) involving ϵ_i is negated,

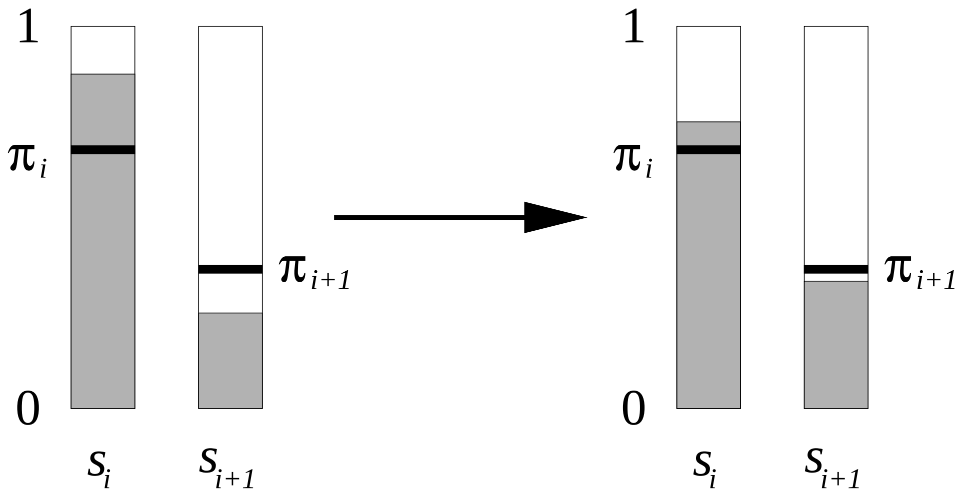

Figure 3.2: A fluid flow intuition: if s_i is too high, and s_{i+1} is too low, there will be flow from s_i to s_{i+1}.

but now ϵ_i is positive. If $\epsilon_{i-1}, \epsilon_i$ and ϵ_{i+1} have mixed signs, this can only decrease the value of the term (3.7). (See Figure 3.2.)

It is simple to check inductively that one can choose an increasing sequence of w_i (starting with $w_0 = 0, w_1 = 1$) and a δ such that the w_i satisfy the above restriction. For example, one can break the terms up into two subsequences. The first subsequence consists of all w_i such that π_i satisfies $\lambda(2\pi_i + 1) \geq \frac{1+\lambda}{2}$. For these i we can choose $w_{i+1} = w_i + \frac{w_i(1-\delta)-w_{i-1}}{3}$. Because this subsequence has only finitely many terms, we can choose a suitably small δ so that this sequence is increasing. For sufficiently large i, we must have $\lambda(2\pi_i + 1) < \frac{1+\lambda}{2} < 1$, and for these i we may set $w_{i+1} = w_i + \frac{2w_i(1-\delta)-2w_{i-1}}{1+\lambda}$. This subsequence of w_i will be increasing for suitably small δ. Also, this sequence is dominated by a geometrically increasing sequence, so the condition $s_j(0) = 0$ for some j is sufficient to guarantee that $\Phi(0) < \infty$.

Comparing terms involving ϵ_i in Φ and $d\Phi/dt$ yields that $d\Phi/dt \leq -\delta\Phi$. Hence $\Phi(t) \leq \Phi(0)e^{-\delta t}$ and thus Φ converges exponentially.

We now consider the technical problem of defining $d\Phi/dt$ when $\epsilon_i(t) = 0$ for some i. Since we are interested in the forward progress of the system, it is sufficient to consider the upper right-hand derivatives of ϵ_i. (See, for instance, [56, p. 16].) That is, we may define

$$\left.\frac{d|\epsilon_i|}{dt}\right|_{t=t_0} \equiv \lim_{t \to t_0^+} \frac{|\epsilon_i(t)|}{t - t_0},$$

and similarly for $d\Phi/dt$. Note that this choice has the following property: if $\epsilon_i(t) = 0$, then $\left.\frac{d|\epsilon_i|}{dt}\right|_{t=t_0} \geq 0$, as it intuitively should be. The above proof applies unchanged with this definition of $d\Phi/dt$, with the understanding that the case $\epsilon_i > 0$ includes the case where $\epsilon_i = 0$ and $d\epsilon_i/dt > 0$, and similarly for the case $\epsilon_i < 0$. ■

Theorem 3.6 yields the following corollary:

Corollary 3.7 *Under the conditions of Theorem 3.6, the L_1-distance from the fixed point $D(t) = \sum_{i=1}^{\infty} |s_i(t) - \pi_i|$ converges exponentially to 0.*

Proof: As the w_i of Theorem 3.6 are all at least 1, $\Phi(t) \geq D(t)$ and the corollary is immediate. ■

Corollary 3.7 shows that the L_1-distance to the fixed point converges exponentially quickly to 0. Hence, from any suitable starting point, the infinite system quickly becomes extremely close to the fixed point. Although it seems somewhat unusual that we had first to prove exponential convergence for a weighted variation of the L_1-distance in order to prove exponential convergence of the L_1-distance, it appears that this approach was necessary; this will be clarified in Section 4.6.

3.3.4 The expected time in the infinite system

Using Theorems 3.4 and 3.6, we now examine the expected time a customer spends in the infinite system.

Corollary 3.8 *The expected time a customer spends in the infinite supermarket system for $d \geq 2$, subject to the condition of Theorem 3.4, converges as $t \to \infty$ to*

$$T_d(\lambda) \equiv \sum_{i=1}^{\infty} \lambda^{\frac{d^i - d}{d-1}}.$$

Furthermore, this number is an upper bound on the expected time in the infinite system for all t when the system is initially empty.

Proof: An incoming customer that arrives at time t becomes the ith customer in the queue with probability $s_{i-1}(t)^d - s_i(t)^d$. Hence the expected time a customer that arrives at time t spends in the system is $\sum_{i=1}^{\infty} i(s_{i-1}(t)^d - s_i(t)^d) = \sum_{i=0}^{\infty} s_i(t)^d$. As $t \to \infty$, by Corollary 3.7, the infinite system converges to the fixed point in the L_1-distance metric. Hence the expected time a customer spends in the system can be made arbitrarily close to $\sum_{i=0}^{\infty} \pi_i^d = \sum_{i=1}^{\infty} \lambda^{\frac{d^i-d}{d-1}}$ for all customers arriving at time $t \geq t_0$ for some sufficiently large t_0, and the result follows. The second result follows since we know that in an initially empty infinite system $s_i(t) \leq \pi_i$ for all t by Theorem 3.4. ■

Recall that $T_1(\lambda) = \frac{1}{1-\lambda}$ from standard queueing theory. Analysis of the summation in Corollary 3.8 reveals the following.

Theorem 3.9 *For $\lambda \in [0,1]$, $T_d(\lambda) \leq c_d(\log T_1(\lambda))$ for some constant c_d dependent only on d. Furthermore,*

$$\lim_{\lambda \to 1^-} \frac{T_d(\lambda)}{\log T_1(\lambda)} = \frac{1}{\log d}.$$

Choosing from $d > 1$ queues hence yields an exponential improvement in the expected time a customer spends in the infinite system, and as $\lambda \to 1^-$ the choice of d affects the time only by a factor of $\log d$. These results are remarkably similar to those for the static case studied in [11] and described in Section 1.2.

Proof: We prove only the limiting statement as $\lambda \to 1^-$; the other statement is proved similarly. Let $\lambda' = \lambda^{1/(d-1)}$. Then

$$T_d(\lambda) = \sum_{i=1}^{\infty} \lambda^{\frac{d^i-d}{d-1}} = \frac{\sum_{i=1}^{\infty} \lambda'^{d^i}}{\lambda^{d/(d-1)}}.$$

Hence

$$\begin{aligned}
\lim_{\lambda' \to 1^-} \frac{T_d(\lambda)}{\log T_1(\lambda)} &= \lim_{\lambda' \to 1^-} \frac{\sum_{i=1}^{\infty} \lambda'^{d^i}}{-\log(1-\lambda)\lambda^{d/(d-1)}} \\
&= \lim_{\lambda' \to 1^-} \frac{\sum_{i=1}^{\infty} \lambda'^{d^i}}{-\log(1-\lambda')} \frac{\log(1-\lambda')}{\log(1-\lambda)} \frac{1}{\lambda^{d/(d-1)}}.
\end{aligned}$$

In the final expression on the right, the last two terms go to 1 as $\lambda \to 1^-$. The result then follows from the following lemma. ■

Lemma 3.10 *Let*

$$F_d(\lambda) = \frac{\sum_{i=0}^{\infty} \lambda^{d^i}}{\log \frac{1}{1-\lambda}}.$$

Then $\lim_{\lambda \to 1^-} F_d(\lambda) = 1/\log d$.

Proof: We show that, for any small enough $\epsilon > 0$, there is a corresponding δ such that for $\lambda > 1 - \delta$,

$$\frac{1}{(\log d) + \epsilon} \leq F_d(\lambda) \leq \frac{1}{(\log d) - \epsilon}.$$

We prove only the left inequality; the right inequality is enitrely similar. We use the following identity:

$$\prod_{i=0}^{\infty} (1 + \lambda^{d^i} + \lambda^{2d^i} + \ldots + \lambda^{(d-1)d^i}) = \frac{1}{1-\lambda}.$$

From this identity it follows that

$$\sum_{i=0}^{\infty} \log(1 + \lambda^{d^i} + \lambda^{2d^i} + \ldots + \lambda^{(d-1)d^i}) = \log \frac{1}{1-\lambda}.$$

For a given ϵ, let $\epsilon' = \epsilon/2$, and let

$$z = \sup \left[\{0\} \cup \{x : 0 < x \leq 1\,,\ \frac{\log(1 + x + x^2 + \ldots x^{d-1})}{x} > \log d + \epsilon'\} \right].$$

Note that $z < 1$. For any fixed λ, we split up the summation in the previous equation to obtain

$$\sum_{i:\lambda^{d^i} < z} \log(1 + \lambda^{d^i} + \ldots + \lambda^{(d-1)d^i}) + \sum_{i:\lambda^{d^i} \geq z} \log(1 + \lambda^{d^i} + \ldots + \lambda^{(d-1)d^i}) = \log \frac{1}{1-\lambda}. \quad (3.8)$$

The leftmost term of equation (3.8) is bounded by a constant, dependent only on z and independent of λ. Hence

$$\begin{aligned} \sum_{i:\lambda^{d^i} < z} \log(1 + \lambda^{d^i} + \ldots + \lambda^{(d-1)d^i}) + \sum_{i:\lambda^{d^i} \geq z} \log(1 + \lambda^{d^i} + \ldots + \lambda^{(d-1)d^i}) \quad \leq \\ c_z + (\log d + \epsilon') \sum_{i:\lambda^{d^i} < z} \lambda^{d^i} + (\log d + \epsilon') \sum_{i:\lambda^{d^i} \geq z} \lambda^{d^i}, \end{aligned} \quad (3.9)$$

where c_z is a constant dependent only on z and is independent of λ. Combining equations (3.8) and (3.9) yields

$$\begin{aligned} (\log d + \epsilon') \sum_{i=0}^{\infty} \lambda^{d^i} + c_z \quad &\geq \quad \log \frac{1}{1-\lambda}\ , \text{ or} \\ F_d(\lambda) + \frac{c_z}{(\log d + \epsilon') \left(\log \frac{1}{1-\lambda} \right)} \quad &\geq \quad \frac{1}{(\log d) + \epsilon'}. \end{aligned}$$

We now choose δ small enough so that for $\lambda > 1-\delta$,

$$\frac{1}{(\log d)+\epsilon'} - \frac{c_z}{(\log d+\epsilon')\left(\log\frac{1}{1-\lambda}\right)} \geq \frac{1}{(\log d)+\epsilon},$$

and the lemma follows. ■

3.4 From infinite to finite: Kurtz's theorem

The supermarket model is an example of a *density dependent family of jump Markov processes*, the formal definition of which we give shortly. Informally, such a family is a one parameter family of Markov processes, where the parameter n corresponds to the total population size (or, in some cases, area or volume). The states can be normalized and interpreted as measuring population densities, so that the transition rates depend only on these densities. As we have seen, in the supermarket model, the transition rates between states depend only upon the densities s_i. Hence the supermarket model fits our informal definition of a density dependent family. The *infinite system* corresponding to a density dependent family is the limiting model as the population size grows arbitrarily large.

Kurtz's work provides a basis for relating the infinite system for a density dependent family to the corresponding finite systems. Essentially, Kurtz's theorem provides a law of large numbers and Chernoff-like bounds for density dependent families. Again, before launching into the technical details, we provide some informal intuition. The primary differences between the infinite system and the finite system are: by

- The infinite system is deterministic; the finite system is random.
- The infinite system is continuous; the finite system has jump sizes that are discrete values.

Imagine starting both systems from the same point for a small period of time. Since the jump rates for both processes are initially the same, they will have nearly the same behavior. Now suppose that if two points are close in the infinite dimensional space then their transition rates are also close; this is called the *Lipschitz condition*, and it is a precondition for Kurtz's theorem. Then even after the two processes separate, if they remain close, they will still have nearly the same behavior. Continuing this process inductively over time, we can bound how far the processes separate over any interval $[0, T]$.

In Section 3.5, we will apply Kurtz's results to the finite supermarket model to obtain bounds on the expected time a customer spends in the system and the maximum queue length.

Theorem 3.11 *For any fixed T, the expected time a customer spends in an initially empty supermarket system of size n over the interval $[0,T]$ is bounded above by*

$$\sum_{i=1}^{\infty} \lambda^{\frac{d^i-d}{d-1}} + o(1),$$

where the $o(1)$ is understood as $n \to \infty$ and may depend on T.

The $o(1)$ term in Theorem 3.11 is the correction for the finite system, while the main term is the expected time in the infinite system from Corollary 3.8. Similarly, one can bound the maximum load:

Theorem 3.12 *For any fixed T, the length of the longest queue in an initially empty supermarket system of size n over the interval $[0,T]$ is $\frac{\log\log n}{\log d} + O(1)$ with high probability, where the $O(1)$ term depends on T and λ.*

In the case where customers have only one choice, in equilibrium the expected time in the system is $\frac{1}{1-\lambda}$ and (as $\pi_i = \lambda^i$) the maximum load is $O(\log n)$. Hence, in comparing the systems where customers have one choice and customers have $d \geq 2$ choices, we see that the second yields an exponential improvement in both the expected time in the system and in the maximum observed load for sufficiently large n. In practice, simulations reveal that this behavior is apparent even for relatively small n over long periods of time, suggesting that the smaller order terms given in the above theorems can be improved. We will discuss this point further in Section 3.5.

3.4.1 Kurtz's theorem

We now give a more technical presentation of Kurtz's theorem. The presentation is based on [52], although we have extended it to include certain infinite dimensional systems.[4] We begin with the definition of a density dependent family of Markov chains, as

[4]Recall that the system is *infinite dimensional* because s_i represents the fraction of servers with load at least i, and the state is the vector $\vec{s} = (s_1, s_2, \ldots)$. An *infinite system* is one where the size of the system, in terms of the number of servers n, goes to infinity. As the epidemic model of Section 3.2 shows, an infinite system need not be infinite dimensional.

in [52, Chapter 8], although we extend the definition to countably many dimensions.[5] For convenience we drop the vector notation where it can be understood by context. Let $\mathbf{Z}^*$ be either $\mathbf{Z}^d$ for some dimension d, or $\mathbf{Z}^{\mathbf{N}}$, as appropriate. Given a set of transitions $L \subseteq \mathbf{Z}^*$ and a collection of nonnegative functions β_l for $l \in L$ defined on a subset $E \subset \mathbf{R}^*$, a *density dependent family of Markov chains* X_n is a sequence $\{X_n\}$ of jump Markov processes such that the state space of X_n is $E_n = E \cap \{n^{-1}k : k \in \mathbf{Z}^*\}$ and the transition rates of X_n are

$$q^{(n)}_{x,y} = n\beta_{n(y-x)}(x), \quad x, y \in E_n.$$

As an example of this definition, consider the supermarket model for $d = 2$ with n queues. The state of the system is $\vec{s} = k/n$, where $\vec{s}$ is given by the vector of the s_i and k is the state scaled to the integers. That is, $\vec{s}$ represents the state by the fraction of servers of size at least i, and k represents the state by the number of servers of size at least i. Note that we may think of the state of the system either as $\vec{s}$ or k, as they are the same except for a scale factor. The possible transitions from k is given by the set $L = \{\pm e_i \ : \ i \geq 1\}$, where the e_i are standard unit vectors; these transitions occur when a customer either enters or departs. The transition rates are given by $q^{(n)}_{k,k+l} = n\beta_l(k/n) = n\beta_l(\vec{s})$, where $\beta_{e_i}(\vec{s}) = \lambda(s_{i-1}^2 - s_i^2)$, and $\beta_{-e_i}(\vec{s}) = s_i - s_{i+1}$. These rates determined our infinite system (3.5).

It follows from [52, Chapter 7], that a Markov process $\hat{X}_n$, with intensities $q^{(n)}_{k,k+l} = n\beta_l(k/n)$ satisfies

$$\hat{X}_n(t) = \hat{X}_n(0) + \sum_{l \in L} l Y_l \left(n \int_0^t \beta_l \left(\frac{\hat{X}_n(u)}{n} \right) du \right),$$

where the $Y_l(x)$ are independent standard Poisson processes. This equation has a natural interpretation: the process at time t is determined by the starting point and the rate of each transition integrated over the history of the process. In the supermarket system, $\hat{X}_n$ is the unscaled process with state space $\mathbf{Z}^{\mathbf{N}}$ that records the number of servers with at least i customers for all i, and $\hat{X}_n(0)$ is the initial state, which we usually take to be the empty system.

[5] This association of finite dimensional and infinite dimensional spaces is a technical convenience we use in this section. A more rigorous approach would embed the infinite dimensional system in an appropriate Banach space, such as the space of sequences with limit 0. Note that, under the conditions of Theorem 3.4, the system state is always a sequence $(s_i)_{i=0}^{\infty}$ with the limit of the s_i being 0, and hence the system lies in this space. See, for example, [27] for this more general treatment.

We set

$$F(x) = \sum_{l \in L} l \beta_l(x), \tag{3.10}$$

and by setting $X_n = n^{-1}\hat{X}_n$ to be the appropriate scaled process, we have from the above:

$$X_n(t) = X_n(0) + \sum_{l \in L} l n^{-1} \tilde{Y}_l \left(n \int_0^t \beta_l(X_n(u)) du \right) + \int_0^t F(X_n(u)) du, \tag{3.11}$$

where $\tilde{Y}_l(x) = Y_l(x) - x$ is the Poisson process centered at its expectation.

As we shall verify shortly, the deterministic limiting process is given by

$$X(t) = x_0 + \int_0^t F(X(u)) du, \; t \geq 0\,, \tag{3.12}$$

where $x_0 = \lim_{n \to \infty} X(0)$. An interpretation relating equations (3.11) and (3.12) is that as $n \to \infty$, the value of the centered Poisson process $\tilde{Y}_l(x)$ will go to 0 by the law of large numbers. In the supermarket model, the deterministic process corresponds exactly to the differential equations we have in system (3.5), as can be seen by taking the derivative of equation (3.12). Also, in the supermarket model we have $x_0 = X_n(0) = (1, 0, 0, \ldots)$ in the case where we begin with the empty system.

We must also introduce a condition that ensures uniqueness for the corresponding limiting deterministic process, the differential equation $\dot{X} = F(X)$. The appropriate condition is that the differential equation be Lipschitz; that is, for some constant M,

$$|F(x) - F(y)| \leq M|x - y|.$$

That this is a sufficient condition for uniqueness in the finite-dimensional case is standard; for the countably infinite dimensional case, a bit more work is required ([26, Theorem 3.2] or [1, p. 188, Theorem 4.1.5]).

We now present Kurtz's theorem. We note that the proof is essentially exactly the same as that given in [52, Chapter 8] or [32, Chapter 11], generalized to the case of countably infinite dimensions.

Theorem 3.13 [Kurtz] *Suppose we have a density dependent family (of possibly countably infinite dimension) satisfying the Lipschitz condition*

$$|F(x) - F(y)| \leq M|x - y|$$

for some constant M. Further suppose $\lim_{n\to\infty} X(0) = x_0$, and let X be the deterministic process:

$$X(t) = x_0 + \int_0^t F(X(u))du,\ t \geq 0.$$

Consider the path $\{X(u) : u \leq t\}$ for some fixed $t \geq 0$, and assume that there exists a neighborhood K around this path satisfying

$$\sum_{l\in L} |l| \sup_{x\in K} \beta_l(x) < \infty. \tag{3.13}$$

Then

$$\lim_{n\to\infty} \sup_{u\leq t} |X_n(u) - X(u)| = 0 \ a.s.$$

Proof: We follow [52, Chapter 8]. Let $\sup_{x\in K} \beta_l(x) = \bar{\beta}_l$. Then

$$\begin{aligned}
\epsilon_n(t) &\equiv \sup_{u\leq t} |X_n(u) - X_n(0) - \int_0^u F(X_n(s))ds| &(3.14)\\
&\leq \sum_{l\in L} |l| n^{-1} \sup_{u\leq t} |\tilde{Y}_l(n\bar{\beta}_l u)| &(3.15)\\
&\leq \sum_{l\in L} |l| n^{-1} (Y_l(n\bar{\beta}_l t) + n\bar{\beta}_l t), &(3.16)
\end{aligned}$$

where (3.15) follows from equation (3.11) and the last inequality is term by term. We can apply the law of large numbers directly to the process on the right, to find:

$$\begin{aligned}
\lim_{n\to\infty} \sum_{l\in L} |l| n^{-1} (Y_l(n\bar{\beta}_l t) + n\bar{\beta}_l t) &= \sum_{l\in L} 2|l|\bar{\beta}_l t\\
&= \sum_{l\in L} \lim_{n\to\infty} |l| n^{-1} (Y_l(n\bar{\beta}_l t) + n\bar{\beta}_l t).
\end{aligned}$$

The last equality uses the condition (3.13) to guarantee that all summations are finite, and hence we can interchange the limit and the summation. Since the inequality from (3.15) to (3.16) is term by term, we can interchange the limit and the summation in (3.15) as well, to obtain:

$$\lim_{n\to\infty} \epsilon_n(t) \leq \sum_{l\in L} \lim_{n\to\infty} |l| n^{-1} \sup_{u\leq t} |\tilde{Y}_l(n\bar{\beta}_l u)|.$$

The right hand side goes to 0 almost surely.

We can derive from the above that for all $u \leq t$,

$$|X_n(u) - X(u)| \leq |X_n(0) - x_0| + \epsilon_n(u) + \int_0^u M|X_n(s) - X(s)|ds. \tag{3.17}$$

We now apply Gronwall's inequality (see [32, p. 498] or [70, p. 78]):

Lemma 3.14 [Gronwall's inequality] *Let $f(t)$ be a bounded function on $[0, T]$ satisfying*

$$f(t) \leq \epsilon + \delta \cdot \int_0^t f(s)ds$$

for $0 \leq t \leq T$, where δ and ϵ are positive constants. Then for $t \in [0, T]$, we have

$$f(t) \leq \epsilon \mathrm{e}^{\delta t}.$$

Applying Gronwall's inequality to equation (3.17), we have

$$|X_n(u) - X(u)| \leq (|X_n(0) - x_0| + \epsilon_n(t))\mathrm{e}^{Mu}. \tag{3.18}$$

The theorem follows. ■

Theorem 3.13 says that the infinite system is the correct limiting process as the system size n goes to infinity. In fact, the theorem yields a great deal more. Using equation (3.18) of Theorem 3.13, we can determine a bound on the deviation between the finite system and the infinite system that holds over bounded time intervals $[0, T]$ with high probability by finding a bound for $\epsilon_n(t)$ that holds with high probability. We use this to bound the deviation of the finite supermarket model from the infinite system in Section 3.5. Since we know that the infinite system follows a trajectory that quickly converges to its fixed point by Corollary 3.7, by applying Kurtz's theorem appropriately we may conclude that over a bounded time interval the finite system follows a trajectory that approaches the fixed point with high probability.

Note that, by equation (3.18), in a straightforward application of Kurtz's theorem, the error bound is exponential in T. Since, by Corollary 3.7, the supermarket model converges exponentially, it would seem that we could get around this problem by using the following technique: given the starting point, choose some time T_0 so that the infinite system is within ϵ of fixed point (in terms of L_1-distance) by time T_0. By the proof of Kurtz's theorem, with very high probability (exponential in n) the finite system should be within ϵ of the infinite system over the interval $[0, T_0]$, and hence within 2ϵ of the fixed point at T_0. Now break the time interval $[T_0, T]$ into blocks of time T_1, where T_1 is the maximum time required to half the distance to the fixed point in the infinite system. Consider the finite system over the interval $[T_0, T_0 + T_1]$. The corresponding infinite system, starting at the same point as the finite system, goes from distance at most 2ϵ away from the fixed point to at most ϵ away from the fixed point. Using Kurtz's theorem, with very high probability (exponential in n) the finite system should be within ϵ of the infinite system over the interval $[T_0, T_0 + T_1]$. Hence the finite system stays within 3ϵ of the fixed point over this whole interval, and end within 2ϵ of the fixed point at $T_0 + T_1$. We can now repeat this argument for the next block of T_1 time units, and so on. This argument can be used to show that once the finite system gets close to the fixed point, it remains close for periods of time that are exponentially (in n) long. Furthermore, since the infinite system converges exponentially, even if the finite system does deviate from the fixed point over some interval of time, it is likely to move back towards the fixed point almost immediately. This argument is the basis for the Freidlen-Wentzell theory, presented for example in [70, Chapter 6].

Unfortunately, we have not been able to formalize this argument for the infinite dimensional supermarket model, because of some technical problems related to working in infinite dimensions. The problem is that T_1, as we have informally defined it, may not exist in the infinite dimensional case. The proof of Theorem 3.6 shows that there is some T_1 such that the value of the potential function $\Phi(t)$ is halved after T_1 units of time, but this T_1 may not half the actual L_1-distance to the fixed point. This setback appears relatively minor, for the following reasons. We note, without proof, that this argument can be formalized for a finite dimensional variation of the supermarket model that we present in Section 4.4.2, which corresponds to a system where a maximum queue size is allowed. (In the finite dimensional case, $\Phi(t)$ and the L_1-distance differ by at most a constant factor, so a suitable T_1 can be found.) A proof for the infinite dimensional case may therefore be possible using a limiting argument based on a sequence of finite dimensional systems. (See, for example, the work of Miller and Michel [57].) For the supermarket model, each system in this sequence would have a bound on the maximum queue size, and we would examine the limit as the maximum allowed queue length increases to infinity.

This main point, however, is that this argument suggests that although the finite system will move far from the fixed point of the infinite system over a suitably long period of time, most of the time it remains near the fixed point. Our simulation results in Section 3.6 verify that this is the case.

3.5 Proofs for finite systems

As indicated earlier, we can apply Kurtz's theorem to the supermarket model to obtain bounds on the expected time a customer spends in the system and the maximum queue length. We first note that the preconditions of Theorem 3.13 hold. The rate at which jumps occur is bounded above by $\lambda + 1$ everywhere. We also need the following:

Lemma 3.15 *The supermarket model satisfies the Lipschitz condition.*

Proof: Let $x = (x_i)$ and $y = (y_i)$ be two states of the supermarket model. Then

$$\begin{aligned} |F(x) - F(y)| &\leq \sum_{i=1}^{\infty} |\lambda(x_{i-1}^d - x_i^d) - (x_i - x_{i+1}) - \lambda(y_{i-1}^d - y_i^d) + (y_i - y_{i+1})| \\ &\leq 2\sum_{i=0}^{\infty} |x_i - y_i| + 2\lambda \sum_{i=0}^{\infty} |x_i^d - y_i^d| \\ &\leq \sum_{i=0}^{\infty} (2 + 2d\lambda)|x_i - y_i|, \end{aligned}$$

where we have used the fact that $0 \leq x_i, y_i \leq 1$ for all i. ■

In order to bound the error between the finite supermarket model with n queues and the infinite system, we need a bound on the quantity $\epsilon_n(t)$ in Theorem 3.13, which we find by using equation (3.15). Equation (3.15) bounds $\epsilon_n(t)$ by a summation, where each term contains a factor that is the maximum deviation of a Poisson process from its mean over the course of the process. We begin by showing the following intuitive lemma: if the Poisson process deviates a large amount from its mean at some point, it is likely to deviate a large amount from its mean at the end of process. For the following lemma, let $\tilde{Z}(t)$ represent a centered Poisson process that has a fixed rate over the interval $[0, t]$.

Lemma 3.16

$$\mathbf{Pr}(\sup_{u \leq t} |\tilde{Z}(t)| \geq 2a) \leq \frac{\mathbf{Pr}(|\tilde{Z}(t)| > a)}{\mathbf{Pr}(|\tilde{Z}(t)| < a)}.$$

Proof: If $\sup_{u \leq t} \tilde{Z}(t) \geq 2a$, then let x be the first time at which $\tilde{Z}(x) = 2a$. The only way that the process can end with $|\tilde{Z}(t)| < a$ is if $|\tilde{Z}(t) - \tilde{Z}(x)| > a$. But the remainder of the process from x to t is independent of its past, so $\tilde{Z}(t) - \tilde{Z}(x)$ is distributed as $\tilde{Z}(t - x)$, and further $\mathbf{Pr}(\tilde{Z}(t - x) > a) \leq \mathbf{Pr}(\tilde{Z}(t) > a)$. Hence

$$\mathbf{Pr}(\sup_{u \leq t} |\tilde{Z}(t)| \geq 2a)\mathbf{Pr}(|\tilde{Z}(t)| < a) \leq \mathbf{Pr}(|\tilde{Z}(t)| > a).$$

■

We have reduced the problem of bounding the maximum deviation of the Poisson process from its mean to bounding the final deviation of the Poisson process from its mean. This latter quantity is more easily bounded by applying Chernoff-type bounds to the Poisson distribution that are similar to those in Lemma 1.3.

Lemma 3.17 [7, Theorem A.15] *Let P have a Poisson distribution with mean μ. For $\epsilon > 0$,*

$$\mathbf{Pr}(P \leq \mu(1-\epsilon)) \leq e^{\epsilon^2 \mu/2} \tag{3.19}$$

$$\mathbf{Pr}(P \geq \mu(1+\epsilon)) \leq \left[e^{\epsilon}(1+\epsilon)^{-(1+\epsilon)}\right]^{\mu}. \tag{3.20}$$

∎

Using these ideas, we can bound $\epsilon_n(t)$ for the supermarket model; we omit the technical proof.

Lemma 3.18 *For the supermarket model, the value of* $\sup_{u \leq t} |X_n(u) - X(u)|$, *which is the maximum L_1-distance between the finite and the infinite process over the time interval* $[0, t]$, *is $O\left(\frac{\log^2 n}{\sqrt{n}}\right)$ with high probability.* ∎

Note that the constant in the $O\left(\frac{\log^2 n}{\sqrt{n}}\right)$ term of Lemma 3.18 may be exponential in t, d, and λ. We now prove the theorems from Section 3.4.

Theorem 3.11 *For any fixed T, the expected time a customer spends in an initially empty supermarket system with $d \geq 2$ over the interval $[0, T]$ is bounded above by*

$$\sum_{i=1}^{\infty} \lambda^{\frac{d^i - d}{d-1}} + o(1),$$

where the $o(1)$ is understood as $n \to \infty$.

Proof: Suppose we start with an empty system. Then by Theorem 3.4 we know that for the infinite system, $s_i(t) \leq \pi_i$ over the entire interval $[0, T]$. Using Lemma 3.18, with high probability the L_1-distance between the finite and infinite systems is $O\left(\frac{\log^2 n}{\sqrt{n}}\right)$. We also note that, with high probability, the maximum queue size in a supermarket system with n queues over the interval $[0, T]$ is $O(\log n)$, based on a comparison with a system of n independent M/M/1 queues, as described in the remark after Lemma 3.1. Hence, with high probability, the total difference in the expected time between the finite and infinite systems is $O\left(\frac{\log^3(n)}{\sqrt{n}}\right)$ with high probability. This term is clearly $o(1)$. In the case where one of the high probability events does not hold, one can again use a comparison with independent M/M/1 queues to show that the expected time in the system is at most $O(\log n)$. Since these events fail to happen with probability $O(\frac{1}{n})$, the additional term one must add to the expected time for this case is also $o(1)$. ∎

From the proof of Kurtz's theorem, the $o(1)$ term of Theorem 3.11 depends exponentially on T, but as we have mentioned, this appears to be an artifact of our proof technique. Similarly, one can bound the maximum load:

Theorem 3.12 *For any fixed T, the length of the longest queue in an initially empty supermarket system with $d \geq 2$ over the interval $[0,T]$ is $\frac{\log\log n}{\log d} + O(1)$ with high probability, where the $O(1)$ term depends on T and λ.*

Proof: The proof has the following sketch: we first show that, with high probability, in the infinite system the fraction of queues with $\frac{\log\log n}{\log d} + c_1$ customers is much smaller than $1/n$ over the entire time period $[0,T]$ for some constant c_1. This implies that with high probability, there are only a small number of queues with at least $\frac{\log\log n}{\log d} + c_1$ customers in the finite system of size n over the interval $[0,T]$. We then conclude by using this fact to show that, with high probability, no queue ever has $\frac{\log\log n}{\log d} + c_2$ customers for some larger constant c_2.

In the infinite system of an initially empty supermarket model, $s_i(t) \leq \pi_i$ for all t by Theorem 3.4. Since $\pi_i = \lambda^{\frac{d^i-1}{d-1}}$, we have $\pi_k < 1/n$ for $k = \frac{\log\log n}{\log d} + c_1$ and some constant c_1. Hence, in the infinite system, $s_k(t) < 1/n$ over the entire interval $[0,T]$. It follows from Theorem 3.13 and Lemma 3.18 that, with high probability, in the finite system $s_k(t) = O\left(\frac{\log^3 n}{\sqrt{n}}\right)$ over the entire interval $[0,T]$.

We now show that, if $s_k(t) = O\left(\frac{\log^3 n}{\sqrt{n}}\right)$ over the interval $[0,T]$, then the maximum queue size over the interval is $k + c_2$ for some constant c_2 with high probability. Note that the total arrival rate into the finite system is λn, and hence that the expected number of customers that enter the system over the time interval is $\lambda n T$. By the Chernoff-type bounds of Lemma 3.17, the number of customers that enter the system is at most $2\lambda n T$ with probability exponentially small in n. We now use the same idea as in Theorem 1.1. The probability that any entering customer chooses d queues of already containing k or more customers is at most $\left(\frac{\alpha \log^3 n}{\sqrt{n}}\right)^d$ for some constant α. Using Chernoff bounds (Lemma 1.3, equation (1.3)), with high probability the number of customers that join a queue with at least k customers is at most polylog(n). Hence the number of queues that ever have at least $k+1$ customers is at most polylog(n). Repeating this argument, it is easy to check that with high probability no queue ever has $k+2$ customers. ■

3.6 Simulation results

We provide the results of some simulations based on the supermarket model.[6] Table 3.1 presents results for a system of $n = 100$ queues at various arrival rates. The results are based on the average of 10 runs, where each run simulates 100,000 time steps, and the first 10,000 time steps are ignored in recording data in order to give the system time to approach equilibrium. For arrival rates of up to 95% of the service rate (i.e., $\lambda = 0.95$), the predictions are within a few percent of the simulation results. Even at 99% of capacity, the prediction from the infinite system is within 10% of the simulations when two queues are selected, and at 99.9% (generally an unreasonably high load on a system), the predictions are off only by factors close to 2. It is not surprising that the error increases as the arrival rate or the number of choices available to a customer increases, as these parameters affect the error term in Kurtz's theorem (for example, they affect the constant used to verify the Lipschitz condition in Lemma 3.15). As one would expect, however, the approximation does improve if the number of queues is increased, as can be seen by the results for 500 queues given in Table 3.2.

The simulations clearly demonstrate the impact of having two choices. Recall that, in equilibrium, the expected time a customer spends in the system given one choice ($d = 1$) is $1/(1-\lambda)$. Hence, as shown in Table 3.2, when $\lambda = 0.99$ the expected time in the system when $d = 1$ is 100.00; with two choices, this drops to under 6. Allowing additional choices leads to much less significant improvements. When the arrival rate is smaller the effect is less dramatic, but still apparent, and when $\lambda = 0.999$, Table 3.2 shows the effect is even more pronounced. The qualitative behaviors that we predicted with our analysis are thus readily observable in our simulations even of relatively small systems. This lends weight to the predictive power of our theoretical results in practical settings.

[6] In these simulations, choices were made without replacement, as this method is more likely to be used in practice. The infinite system is the same regardless of whether the choices are made with or without replacement; we have described the model with replacement to simplify the exposition. The same holds for other simulation results in this thesis unless explicitly noted.

Choices	λ	Simulation	Prediction	Relative Error (%)
2	0.50	1.2673	1.2657	0.1289
	0.70	1.6202	1.6145	0.3571
	0.80	1.9585	1.9475	0.5742
	0.90	2.6454	2.6141	1.1981
	0.95	3.4610	3.3830	2.3028
	0.99	5.9275	5.4320	9.1227
	0.999	14.0790	8.6516	62.7328
3	0.50	1.1277	1.1252	0.2146
	0.70	1.3634	1.3568	0.4858
	0.80	1.5940	1.5809	0.8314
	0.90	2.0614	2.0279	1.6533
	0.95	2.6137	2.5351	3.1002
	0.99	4.4080	3.8578	14.2607
	0.999	11.7193	5.9021	98.5593
5	0.50	1.0340	1.0312	0.2637
	0.70	1.1766	1.1681	0.7250
	0.80	1.3419	1.3289	0.9789
	0.90	1.6714	1.6329	2.3564
	0.95	2.0730	1.9888	4.2363
	0.99	3.4728	2.9017	19.6825
	0.999	9.3366	4.3001	117.1240

Table 3.1: Simulations versus estimates for the supermarket model: 100 queues.

Choices	λ	Simulation	Prediction	Relative Error (%)
1	0.99	–	100.00	–
2	0.99	5.5413	5.4320	2.0121
3	0.99	3.9518	3.8578	2.4366
5	0.99	3.0012	2.9017	3.4305
1	0.999	–	1000.00	–
2	0.999	9.3994	8.6516	8.6435
3	0.999	7.0497	5.9021	19.4428
5	0.999	5.5801	4.3001	29.7665

Table 3.2: Simulations versus estimates for the supermarket model: 500 queues.

Chapter 4

Infinite systems for other load balancing problems

4.1 Introduction

In the last chapter we introduced a new methodology for studying load balancing problems, based upon the idea of the *infinite system*. The technique required the following steps:

- Set up the appropriate infinite system – a system of differential equations.
- Study the behavior of the infinite system, either numerically or by proving convergence to a fixed point.
- Deduce the finite system behavior from that of the infinite system using Kurtz's theorem.

The example of the supermarket model suggests that the infinite system approach can lead to relatively straightforward analyses of complicated systems once the appropriate differential equations have been established. In this chapter, we expand upon this methodology by applying it to a variety of other simple randomized load balancing schemes, many of which have so far resisted analysis. One goal of this chapter is to demonstrate the advantages of this approach: simplicity, generality, and accuracy. The second goal is to gain further insight into practical aspects of load balancing by studying other load balancing strategies and their variations.

Because the emphasis of this chapter is the infinite system approach, we will primarily focus on developing the infinite systems, and set aside questions of convergence until late in the chapter. Also, we will not explicitly use Kurtz's theorem for each system we study; instead, we simply determine the infinite systems, keeping in mind the global principle that Kurtz's theorem can be used to justify the connection between the infinite and finite systems. Of course the preconditions to Kurtz's theorem (Theorem 3.13) must be checked, as was done for the supermarket system in Section 3.5: the jump rates must be bounded, and the infinite system must satisfy the Lipschitz condition. All of the systems we study satisfy these preconditions. Kurtz's theorem allows one to conclude that large finite systems are expected to stay close to the infinite system for small enough intervals of time, and as we have seen for the supermarket model in Section 3.5, this can be used to derive bounds on finite system performance.

We begin by applying the infinite system approach to the GREEDY strategy studied by Azar, Broder, Karlin, and Upfal in the static case [11]. We shall derive an alternative proof of the $\log\log n/\log d + O(1)$ upper bound on the maximum load in the static case, and also demonstrate that the infinite system approach provides accurate predictions of actual performance in this case.

We then return to dynamic systems, beginning with generalized version of the supermarket model where the service times may not be exponentially distributed, and the arrival times may not be Poisson. These assumptions that we made in Chapter 3 limit the applicability of our results, as in practice arrival and service distributions may not be so simple. We shall focus on the specific example of constant service times, although our approach can be applied to other distributions as well. As an interesting corollary, we show that switching from exponential service times to constant service times with the same mean improves the performance of the supermarket system, in terms of the expected time a customer spends in the infinite system.

We then proceed to study a variety of other load balancing models, including variations of the supermarket model as well as other strategies. Two particularly interesting models involve *thresholds*, in which a customer only makes additional choices if the previous destination choices are too heavily loaded, and *load stealing*, in which tasks do not distribute themselves upon arrival, but instead processors seek out work when they are underutilized. Besides demonstrating that our technique applies to a variety of models and problems, these examples emphasize that even a small amount of additional communication beyond simple random selection can lead to dramatic improvements in performance. Our results for the infinite system are again compared with simulations, which verify the accuracy of our approach.

In Section 4.6, we discuss the convergence of the infinite systems we examine in this chapter. In most cases small variations on the attack we developed in Theorem 3.6 to show exponential convergence of the supermarket model suffice to demonstrate exponential convergence of these other models as well. Rather than prove convergence for each model, we prove a general theorem that applies to many of the systems we consider. In cases where we cannot prove exponential convergence, we can often prove a weaker result; namely, that the fixed point is stable. We also discuss why the potential functions we use do not seem to lead to exponential convergence for some models.

We conclude with a brief discussion of open problems and future directions for this work.

4.2 The static model

We now demonstrate the applicability of the infinite system approach to static problems by returning to the original balls and bins setting described in Chapters 1 and 2. Recall the scenario of the GREEDY strategy of [11] that we presented in Section 1.2: in the case where $m = n$, we begin with n balls and n bins. Balls arrive sequentially, and upon arrival, each ball chooses d bins independently and uniformly at random (with replacement); the ball is then placed in the least loaded of these bins (with ties broken arbitrarily). With high probability, the maximum load is $\log\log n / \log d + O(1)$.

Suppose that instead of the maximum load, we wish to know how many bins remain empty after the protocol GREEDY(d) terminates. This question has a natural interpretation in the task-processor model: how many of our processors are not utilized? The question can also be seen as a matching problem on random bipartite graphs: given a bipartite graph with n vertices on each side such that each vertex on the left has d edges to vertices chosen independently and uniformly at random on the right, what is the expected size of the greedy matching obtained by sequentially matching vertices on the left to a random unmatched neighbor? Our attack, again, is to consider this system as $n \to \infty$. This question has been previously solved in the limiting case as $n \to \infty$ by Hajek using similar techniques [38]. We shall begin by briefly repeating his argument with some additional insights. Once we show

how to answer the question of the number of empty bins, we shall extend it to the more general load balancing problem.

4.2.1 The empty bins problem

To set up the problem as a density dependent Markov chain, we first establish a concept of time. We let t be the time at which exactly $x(t) = nt$ balls have been thrown, and we let $y(t)$ be the fraction of non-empty bins. Then at time t, the probability that a ball finds at least one empty bin among its d choices is $1 - y^d$, and hence we have

$$\frac{dy}{dt} = 1 - y^d. \tag{4.1}$$

A marked difference between the static problem and the supermarket model is that in the static case we are only interested in the progress of the process over a fixed time interval, while in the dynamic case we are interested in the behavior of the model over an arbitrary period of time. In this respect, the static problem is easier than the corresponding dynamic problem.

Theorem 4.1 *Suppose cn balls are thrown into n bins according to the* GREEDY*(d) protocol for some constant c. Let Y_{cn} be the number of non-empty bins when the process terminates. Then* $\lim_{n\to\infty} \mathsf{E}[\frac{Y_{cn}}{n}] = y_c$, *where $y_c < 1$ satisfies*

$$c = \sum_{i=0}^{\infty} \frac{y_c^{id+1}}{(id+1)}.$$

Proof: The preconditions for Kurtz's theorem (Theorem 3.13) are easily checked for the one-dimensional system described by (4.1), so by Kurtz's theorem we have that this differential equation is the correct limiting process.[1] Instead of solving (4.1) for y in terms of t, we solve for t in terms of y: $\frac{dt}{dy} = \frac{1}{1-y^d} = \sum_{i=0}^{\infty} y^{id}$. We integrate, yielding

$$t_0 = \sum_{i=0}^{\infty} \frac{y(t_0)^{id+1}}{(id+1)}. \tag{4.2}$$

From equation (4.2), given d we can solve for $y(t_0)$ for any value of t_0 using for example binary search.[2] In particular, when $t_0 = c$, all of the balls have been thrown,

[1]It appears that there might be a problem here since we consider events occurring at discrete time steps, instead of according to random times from a Poisson process. One can always adopt the convention that each discrete time step corresponds to an amount of time given by an exponentially distributed random variable. In the limiting case, this distinction disappears.

[2]One could also attempt to directly find an equation for y in terms of d and c. Standard integral tables give such equations when $d = 2, 3$ and 4, for example [19].

and the process terminates. Plugging $t_0 = c$ into equation (4.2) yields the theorem, with $y_c = y(c)$. ■

We may actually use the proof of Theorem 3.13 to obtain a concentration result.

Theorem 4.2 *In the notation of Theorem 4.1,* $|\frac{Y_{cn}}{n} - y_c|$ *is* $O\left(\sqrt{\frac{\log n}{n}}\right)$ *with high probability, where the constant depends on* c*.*

Proof: From the proof of Theorem 4.1, we know that y_c is the correct limiting value of $\frac{Y_{cn}}{n}$. By equations (3.18) and (3.15), we obtain a bound for $\epsilon_n(c)$ as the supremum of the deviation of a Poisson process with rate 1 from its expectation over the course of the process. By Lemma 3.16 it suffices to consider the deviation from the mean at the end of the process. Using Lemma 3.17, one may check that this deviation is $O\left(\sqrt{\frac{\log n}{n}}\right)$ with high probability. ■

One can also show that Y_{cn} is close to its mean with high probability using martingale arguments and the method of bounded differences. In the following application of this alternative technique, we assume familiarity with basic martingale theory; see, for example, [7, Chapter 7] for more information. We use the following form of the martingale tail inequality due to Azuma [14]:

Lemma 4.3 **[Azuma]** *Let* $X_0, X_1, \ldots X_m$ *be a martingale sequence such that for each* k*,*

$$|X_k - X_{k-1}| \leq 1.$$

Then for any $\alpha > 0$*,*

$$\mathbf{Pr}(|X_m - X_0| > \alpha\sqrt{m}) < 2\mathrm{e}^{-\alpha^2/2}.$$

Theorem 4.4 *In the notation of Theorem 4.1,* $\mathbf{Pr}(|Y_{cn} - \mathsf{E}[Y_{cn}]| > \alpha\sqrt{cn}) < 2\mathrm{e}^{-\alpha^2/2}$ *for any* $\alpha > 0$*.*

Proof: The argument we present is based on a similar argument presented in [41, Theorem 2]. For $0 \leq j \leq cn$, let $\mathcal{F}_j$ be the σ-field of events corresponding to the possible states after j balls have been placed, and $Z_j = \mathsf{E}[Y_{cn}|\mathcal{F}_j]$ be the associated conditional expectation of Y_{cn}. Then the random variables $\{Z_j\}_{j=0}^{cn}$ form a Doob martingale, and it is clear that

$|Z_j - Z_{j-1}| \leq 1$. The theorem now follows from Lemma 4.3. ■

Theorem 4.4 implies that Y_{cn} is within $O(\sqrt{n \log n})$ of its expected value with high probability. Unlike the infinite system method, however, the martingale approach does not immediately lead us to the value to which Y_{cn}/n converges. This is a standard limitation of the martingale approach: one may prove concentration with no knowledge of the actual mean. This infinite system approach, in contrast, often allows one to find the mean as well as prove concentration around the mean.

4.2.2 Bins with fixed load

We can extend the previous analysis to find the fraction of bins with load k for any constant k as $n \to \infty$. We first establish the appropriate density dependent Markov chain. Let $s_i(t)$ be the fraction of bins with load at least i at time t, where again at time t exactly nt balls have been thrown. Then the corresponding differential equations regarding the growth of the s_i (for $i \geq 1$) are easily determined:

$$\begin{cases} \dfrac{ds_i}{dt} &= (s_{i-1}^d - s_i^d) \quad \text{for} \quad i \geq 1\,; \\ s_0 &= 1. \end{cases} \tag{4.3}$$

The differential equations (similar to (3.5) for the supermarket model) have the following simple interpretation: for there to be an increase in the number of bins with at least i balls, the d choices of a ball about to be placed must all be bins with load at least $i-1$, but not all bins with load at least i.

In contrast to Section 4.2.1, where we could derive a formula for the fraction of empty bins, we are not aware of how to determine explicit formulae for $s_i(t)$ in general. These systems of differential equations can be solved numerically using standard methods, however; for up to any fixed k and t, we can accurately determine $s_k(t)$. By applying Kurtz's theorem (as in Theorem 4.2) or martingale arguments (as in Theorem 4.4) one can show that these results will be accurate with high probability.

We also demonstrate that our technique accurately predicts the behavior of the GREEDY(d) algorithm by comparing with simulation results. The first and third columns of Table 4.1 shows the predicted values of s_i for $d = 2$ and $d = 3$. From these results with $d = 2$, one would not expect to see bins with load five until billions of balls have been

	$d=2$ Prediction	1 million Simulation	$d=3$ Prediction	1 million Simulation
s_1	0.7616	0.7616	0.8231	0.8230
s_2	0.2295	0.2295	0.1765	0.1765
s_3	0.0089	0.0089	0.00051	0.00051
s_4	0.000006	0.000007	$< 10^{-11}$	0
s_5	$< 10^{-11}$	0	$< 10^{-11}$	0

Table 4.1: Predicted behavior for GREEDY(d) and average results from 100 simulations with 1 million balls.

thrown. Similarly, choosing $d = 3$ one expects a maximum load of three until billions of balls have been thrown. These results match those we have presented earlier in Table 2.1, as well as simulation results presented in [10]. We also present the averages from one hundred simulations of one million balls for $d = 2$ and $d = 3$, which further demonstrate the accuracy of the technique in predicting the behavior of the system. This accuracy is a marked advantage of this approach; previous techniques have not provided ways of concretely predicting actual performance.

4.2.3 A new proof of $O(\log \log n)$ bounds

We can also use the above approach to give an alternative proof of the upper bounds on the maximum load of GREEDY(d). Our proof is derived from Theorem 1.1; however, we feel that the approach of looking at the underlying differential equations emphasizes the key features of the proof. The differential equations provide the insight that the s_k decrease quadratically at each level, and hence overall the s_k are doubly exponentially decreasing.

Theorem 4.5 [Azar *et al.* [11]] *Suppose that n balls are thrown into n buckets by the* GREEDY *process. Then the final load is* $\log \log n / \log d + O(1)$ *with high probability.*

Proof: We wish to know the values of $s_i(1)$ in the finite system. Because the s_i are all non-decreasing over time and non-negative, in the infinite system we have from (4.3)

$$\frac{ds_i}{dt} = s_{i-1}^d - s_i^d \leq [s_{i-1}(1)^d]$$

for all $t \leq 1$ and hence

$$s_i(1) \leq [s_{i-1}(1)]^d. \qquad (4.4)$$

It is easy to check that $s_1(1) < 1$ in the infinite system, since a constant fraction of the bins will not be chosen by any of the balls. Hence, there exists a constant c_1 such that for $D = \log\log n / \log d + c_1$ we have $s_D(1) < 1/n$ in the infinite system. We now apply the proof of Theorem 3.13 in this D-dimensional space. It is easy to check that the error term introduced in the proof of Theorem 3.13 (from the $\epsilon_n(1)$) between the finite and the infinite dimensional system is $O(\frac{\text{polylog}(n)}{\sqrt{n}})$ with high probability, as in the proof of Theorem 3.11. Hence, with high probability, the number of bins with load at least D is bounded by $O(n^{2/3})$. At this point we must handle the small tail explicitly, as is done in the proof of Theorem 1.1, using Chernoff bounds. It is simple to verify that with high probability that there are at most $O(n^{1/3})$ bins with load at least $D+1$, $O(\log n)$ bins with load at least $D+2$, and no bins with load at least $D+3$. ∎

The above proof essentially mimics that of Theorem 1.1, with the inductive use of Chernoff bounds replaced by an invocation of Kurtz's more general theorem. One can adapt the lower bound proof in [11] in a similar fashion. Note, however, that the proof cannot be extended for the general case of m balls and n bins, unless $m = cn$ for some constant c. When $m = cn$, the infinite process runs until time c; if m is not a linear function of n, the time until the process terminates is dependent on n, and Kurtz's theorem cannot be applied.

4.3 Constant service times

We now wish to show how to apply our techniques to other service and arrival distributions. The assumptions we have made in studying the supermarket model in the previous chapter, namely that the arrival process is Poisson and that the service times are exponentially distributed, do not accurately describe many (and probably most) real systems, although they are standard in much of queueing theory, because they lead to simple Markovian systems. In this section, we demonstrate how to modify our approach in order to apply it to more general service and arrival distributions. The technique is most easily described by fixing on a specific example. Here we examine the supermarket model where the service time is a fixed constant, 1, for all customers. This model is often more realistic in computer systems, where all jobs may require a fixed amount of service.

Once service times are a fixed constant, the supermarket system with our standard

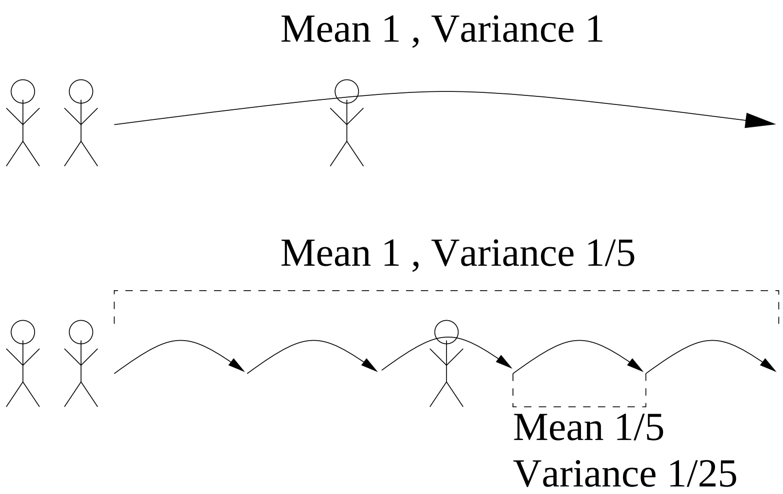

Figure 4.1: Gamma distributed service times with $r = 5$. If one exponentially distributed stage of service is replaced by five shorter exponentially distributed stages, then the variance of the service time is reduced by a factor of five.

state space is no longer Markovian, as the time until the next customer departs depends upon the times at which all customer undergoing service began being served. To use the infinite system technique, we approximate this non-Markovian system by a Markovian one. The approach we use is based on *Erlang's method of stages*, which we shall describe briefly here. Kleinrock's excellent text provides a more detailed explanation [46, Sections 4.2 and 4.3], and Kelly uses this approach in a similar fashion [45, Section 3.3]. We approximate the constant service time with a *gamma distribution*: a single service will consists of r stages of service, where each stage is exponentially distributed with mean $1/r$. As r becomes large, the expected time spent in these r stages of service remains 1 while the variance falls like $1/r$; that is, the total time to complete the r stages behaves like a constant random variable in the limit as $r \to \infty$. The appropriate picture of the process is that we have replaced the single, long exponential server at each queue with r shorter exponential servers in series; note that only a single customer, the one receiving service, can be anywhere inside this group of r stages at one time. (See Figure 4.1.)

The state of a queue will now be the total number of stages remaining that the queue has to process, rather than the number of customers; that is, the state of a queue is

$$r(\text{\# of waiting customers}) + \text{stages of the customer being served}.$$

In practice, since r determines the size of the state space, numerical calculations will be easier if we choose r to be a reasonably small finite number. Our simulations, which appear later in this section, suggest that for $r \approx 20$ the approximations are quite accurate.

There is some ambiguity now in the meaning of a customer choosing the shortest queue. If the number of customers in two queues are the same, can an incoming customer distinguish which queue has fewer stages of service remaining? The two possible answers to this question lead to different systems. Let us first consider the case where we have *aware* incoming customers, who can tell how many stages are left for each of their d choices, and hence choose the best queue when there is a tie in terms of the number of customers. Let s_j be the fraction of queues with at least j stages left to process. For notational convenience we adopt the convention that $s_j = 1$ whenever $j \leq 0$. Then s_j increases whenever an arrival comes to a queue with at least $j - r$ and fewer than j stages left to complete. Similarly, s_j decreases whenever a queue with j stages completes a stage, which happens at rate r. The corresponding system of differential equations is thus

$$\frac{ds_j}{dt} = \lambda(s_{j-r}^d - s_j^d) - r(s_j - s_{j+1}).$$

(Note that in the case where $r = 1$, this corresponds exactly to the standard supermarket model, as one would expect.)

As for the supermarket model, we can identify a unique fixed point $\vec{\pi}$ for this system (with a finite expected number of customers per queue). At the fixed point, $\pi_1 = \lambda$ (intuitively because the arrival rate and exit rate of customers must be equal), and $\pi_i = 1$ for $i \leq 0$. From these initial conditions one can find successive values of π_j from the recurrence

$$\pi_{j+1} = \pi_j - \frac{\lambda(\pi_{j-r}^d - \pi_j^d)}{r}. \tag{4.5}$$

Unfortunately, we have not found a convenient closed form for π_j.

We say that the system has *unaware* customers if the customers can only determine the size of the queue, and not the remaining amount of service required. In a system with unaware customers, if there is a tie for the fewest customers among the queues chosen by an incoming customer, then the customer will choose randomly from those queues with the smallest number of customers. The differential equations are slightly more complicated than in the aware case. Again, let s_j be the fraction of queues with at least j stages left to process. For notational convenience, let $S_i = s_{(i-1)r+1}$ be the fraction of queues with at least i customers, and let $\phi(j) = \lceil \frac{j}{r} \rceil$ be the number of customers in a queue with j stages left to process. There are now two cases to consider in determining when s_j increases, corresponding to whether the shortest queue an incoming customer chooses has $\phi(j)-1$ or $\phi(j)$ customers. Let us consider the first case. The probability that the shortest queue chosen by an incoming customer has $\phi(j)-1$ customers is $S^d_{\phi(j)-1}-S^d_{\phi(j)}$. In this case, s_j will increase only when the incoming customer chooses a queue with at least s_{j-r} stages. Conditioned on the assumption that shortest queue chosen by an incoming customer has $\phi(j)-1$ customers already waiting, the probability that the customer chooses a queue with at least $j-r$ stages is $\frac{s_{j-r}-S_{\phi(j)}}{S_{\phi(j)-1}-S_{\phi(j)}}$. The second case is similar. The corresponding differential equations are

$$\frac{ds_j}{dt}=\lambda(S^d_{\phi(j)-1}-S^d_{\phi(j)})\frac{s_{j-r}-S_{\phi(j)}}{S_{\phi(j)-1}-S_{\phi(j)}}+\lambda(S^d_{\phi(j)}-S^d_{\phi(j)+1})\frac{S_{\phi(j)}-s_j}{S_{\phi(j)}-S_{\phi(j)+1}}-r(s_j-s_{j+1}).$$

Note that the fixed point cannot be determined by a simple recurrence, as the derivative of s_j depends on $S_{\phi(j)}$,$S_{\phi(j)-1}$, and $S_{\phi(j)+1}$. It turns out that the system converges quickly, and hence one can find the fixed point to a suitable degree of accuracy by standard methods, such as simulating the differential equations for a small period of time, or relaxation.

4.3.1 Constant versus exponential service times

One interesting application of the above differential equations is to show that, at the fixed points of the corresponding infinite models, constant service times are better than exponential service times, measured in terms of the expected time a customer spends in the system in equilibrium. More specifically, we show that, if the arrivals form a Poisson process, the fraction of servers with at least k customers is greater when service times are exponential than when service times are constant with the same mean. Although this may not appear surprising, it is far from trivial: a simple variation of a counterexample given by Ross [65] shows that for certain arrival processes, the expected time in systems where customers choose the shortest queue can increase when one changes service times from exponential to constant.

In fact, the question of whether constant service times reduce the expected delay in comparison to exponential service times in a network, and the more general question of whether variance in service times necessarily increases the expected delay of a network, often arises when one tries to use standard queueing theory results to find performance bounds on networks. (See, for example, [39, 59, 60, 64, 71].) Generally, results comparing the two types of systems are achieved using stochastic comparison techniques. Here, we instead compare the fixed points of the corresponding infinite systems.

Let us begin with the case of aware customers where service times have a gamma distribution corresponding to r stages. Recall that the fixed point was given by the recurrence (4.5) as $\pi_{j+1} = \pi_j - \lambda(\pi_{j-r}^d - \pi_j^d)/r$, with $\pi_1 = \lambda$ and $\pi_i = 1$ for $i \leq 0$. The fixed point for the standard supermarket model, as found in Lemma 3.2, satisfies $\pi_{i+1} = \lambda\pi_i^d$. Since π_1 is λ in both the standard supermarket model and the model with gamma distributed service times, to show that the tails are larger in the standard supermarket model, it suffices to show that $\pi_{\phi(j)+1} \leq \lambda\pi_{\phi(j)}^d$ in the aware customer model. Inductively it is easy to show the following stronger fact:

Theorem 4.6 *In the system with aware customers, for $j \geq 1$,*

$$\pi_j = \frac{\lambda}{r}\sum_{i=j-r}^{j-1} \pi_i^d.$$

Proof: The equality can easily be verified for $1 \leq j \leq r$. For $j > r$, the following induction yields the theorem:

$$\begin{aligned}
\pi_j &= \pi_{j-1} - \frac{\lambda}{r}(\pi_{j-r-1}^d - \pi_{j-1}^d) \\
&= \pi_{j-2} - \frac{\lambda}{r}(\pi_{j-r-1}^d + \pi_{j-r-2}^d - \pi_{j-1}^d - \pi_{j-2}^d) \\
&\vdots \\
&= \pi_{j-r} - \frac{\lambda}{r}\left(\sum_{i=j-2r}^{j-r-1} \pi_i^d - \sum_{k=j-r}^{j-1} \pi_k^d\right) \\
&= \frac{\lambda}{r}\sum_{k=j-r}^{j-1} \pi_k^d.
\end{aligned}$$

Here the last step follows from the inductive hypothesis, and all other steps follow from the recurrence equation (4.5) for the fixed point. ■

We may conclude that the expected time when service times have a gamma distribution with r stages is better than when service times are exponential in the corresponding infinite systems. Since, intuitively, the limit as r goes to infinity corresponds to constant service times, we would like to conclude that the expected time in the system when service times are constant is better than when the service times are exponential in the corresponding infinite systems. A full proof of this statement appears to require handling several technical details and has not been completed. In the sequel, we sketch steps that would be necessary for such a proof.

First, even in a finite system, it is not clear that the limiting distribution as $r \to \infty$ converges to the same distribution as when service times are constant. Indeed, even this step is non-trivial, although a similar proof for networks of quasi-reversible queues by Barbour appears to apply [15]. (See also [45, p.77-78] for more background for this problem.) Now, let $\mathcal{D}_{r,n}$ be the distribution when there are n queues and service times have a gamma distribution of r stages. We must also show that

$$\lim_{r\to\infty} \lim_{n\to\infty} \mathcal{D}_{r,n} = \lim_{n\to\infty} \lim_{r\to\infty} \mathcal{D}_{r,n}.$$

The left hand side is the limit of the infinite systems as the service times approaches a fixed constant; the right hand side is the infinite system when service times are constant, assuming that the limiting distribution as $r \to \infty$ does converge to the same distribution as when service times are constant. Showing that this interchange of limits is justified can be extremely difficult; for example, see Chapter 14 of the book by Shwartz and Weiss [70, pp. 400-405] for a partial justification of such an interchange in a model related to the Aloha protocol.

Because of the great technical difficulties of a formal proof, we only provide the above informal justification that our comparison of exponential and gamma distributed service times in the infinite system can be extended to compare exponential and constant service times by taking the limit as $r \to \infty$.

A similar result holds even in the case of *unaware* customers. The proof is simpler with the following notation: let q_i be the probability than an arriving customer joins a queue with i or more stages to complete at the fixed point of the system with unaware customers. We set $q_i = 1$ for $i \leq 0$. Then along with the fact that $\pi_1 = \lambda$ and $\pi_i = 1$ for $i \leq 0$, the following recurrence describes the fixed point for the unaware system:

$$\pi_{j+1} = \pi_j - \frac{\lambda(q_{j-r} - q_j)}{r}. \tag{4.6}$$

Although we do not know the q_j, we can use this recurrence to compare the standard supermarket model with the unaware model. As noted before Theorem 4.6, it suffices to show that in the unaware model, $\pi_{\phi(j)+1} \leq \lambda \pi^d_{\phi(j)}$.

Theorem 4.7 *At the fixed point in the model with unaware customers, for* $j \geq 1$*,*

$$\pi_j = \frac{\lambda}{r} \sum_{i=j-r}^{j-1} q_i.$$

In particular,

$$\pi_{\phi(j)+1} \leq \lambda \pi^d_{\phi(j)}.$$

Proof: The equality can easily be verified for $1 \leq j \leq r$. For $j > r$, the following induction yields the theorem:

$$\begin{aligned}
\pi_j &= \pi_{j-1} - \frac{\lambda}{r}(q_{j-r-1} - q_{j-1}) \\
&= \pi_{j-2} - \frac{\lambda}{r}(q_{j-r-1} + q_{j-r-2} - q_{j-1} - q_{j-2}) \\
&\vdots \\
&= \pi_{j-r} - \frac{\lambda}{r}\left(\sum_{i=j-2r}^{j-r-1} q_i - \sum_{k=j-r}^{j-1} q_k\right) \\
&= \frac{\lambda}{r} \sum_{k=j-r}^{j-1} q_k
\end{aligned}$$

Here the last step follows from the inductive hypothesis, and all other steps follow from the recurrence equation for the fixed point (4.6). The last line of the theorem now follows by noting that the q_i are decreasing and that $q_{\phi(j)} = \pi^d_{\phi(j)}$. ■

Theorem 4.6 can be used to show that constant service times are better than exponential service times in sufficiently large finite systems as well; however, a formal statement would require showing that the trajectories converge to the fixed point when service times are gamma distributed, which we leave to Section 4.6. Here we note that Theorems 4.6 and 4.7 provide evidence that constant service times may be better than exponential service times regardless of the number of queues, and hence suggest that a more straightforward stochastic comparison argument might be possible for this problem. If such an argument exists, it would likely hold for any values of d and n, and therefore be a much stronger result. (Note that we implicitly assume d is a constant in order to obtain the infinite system, and we have really only proven the result for the infinite systems.) We strongly suspect that a comparison argument is possible; this remains an interesting open question.

λ	Simulation	$r = 10$	$r = 20$	$r = 30$
0.50	1.1352	1.1478	1.1412	1.1390
0.70	1.3070	1.3355	1.3200	1.3148
0.80	1.4654	1.5090	1.4847	1.4766
0.90	1.7788	1.8492	1.8065	1.7923
0.95	2.1427	2.2355	2.1714	2.1500
0.99	3.2678	3.2461	3.1243	3.0644

Table 4.2: Simulations versus estimates for constant service times: 100 queues.

4.3.2 Simulations

To provide evidence for our claim that small values for the number of stages r can be used to achieve good approximations for constant service times, we briefly compare our analytic results with corresponding simulation results. Table 4.2 compares the value of the expected time a customer spends in an infinite system with unaware customers and $d = 2$ choices per customer obtained using various values of r against the results from simulations with constant service times for 100 queues. The simulation results are the average of ten runs, each for 100,000 time units, with the first 10,000 time units excluded to account for the fact that the system begins empty.

As one might expect, the expected time in the system decreases as r increases (and hence as the variance decreases). The results for $r = 20$ lie within 1-2% of the value calculated from the infinite system for all values of λ presented except for $\lambda = 0.99$. We expect that for $\lambda = 0.99$ the infinite system only becomes accurate for a larger number of queues, and hence the discrepancy is not surprising.

4.3.3 Other service times

In principle, this approach could be used to develop deterministic differential equations that approximate the behavior of any service time distribution. This follows from the fact that the distribution function of any positive random variable can be approximated arbitrarily closely by a mixture of countably many gamma distributions [45, Lemma 3.9 and Exercise 3.3.3]. (An interesting discussion of this fact is given in [45, Section 3.3], where Erlang's method is used in a manner similar to that here.) In fact, the service time for each customer can be taken to be distributed as a gamma distribution described above with some number of stages r, each with the same mean $1/m$, where the value of r varies from customer to customer and is chosen according to some fixed probability distribution. Using this fact, we briefly describe a suitable state space for a Markov process that approximates the underlying non-Markovian process. The state of a queue will correspond to a triple (i, j, k). The first coordinate, i, is the number of customers in the queue. The second coordinate, j, is the total number of stages to be completed by the customer currently obtaining service in the queue. The third coordinate, k, is the number of stages the customer currently obtaining service has completed. The resulting process is Markovian and one can now write differential equations describing the behavior of the system in terms of the transition rates between states. Various arrival processes can be handled similarly.

In practice, for the solution of this problem to be computable in a reasonable amount of time, one would need to guarantee that both the number of distributions in the mixture and the number of stages for each distribution are small in order to keep the total number of states reasonably small. Thus the quality of the approximation will depend on how well the service distribution can be approximated by a mixture satisfying these conditions. Although these limitations appear severe, many service distributions can still be handled easily. For example, as we have seen, in the case of constant service times one only needs to use a single gamma distribution with a reasonable number of stages to get a very good approximation. Distributions where the service time takes on one of a small finite number of values can be handled similarly.

4.4 Other dynamic models

In this section, we shall develop infinite systems for a variety of other load balancing models. We shall begin with variations on the supermarket model, and then examine other load balancing strategies. This section is devoted to the development of the infinite systems; questions of convergence and comparisons with simulations will be handled in subsequent sections.

Unless otherwise noted, the system state is represented by a vector $(s_0, s_1, \ldots)$, where s_i is the fraction of queues with at least i customers, as in the supermarket model. Similarly, $(\pi_0, \pi_1, \ldots)$ will refer to the fixed point. We will not explicitly check that these systems are stable, in that the expected number of customers per queue is finite; this is generally straightforward. Also, technically a system may have many fixed points; however, the systems we examine generally have a unique fixed point where the average number of customers per queue is finite, and we shall simply refer to this fixed point as *the* fixed point.

4.4.1 Customer types

One possible way of extending the supermarket model is to allow different customers different numbers of choices. The more choices one gets, the less time one is expected to wait. For example, suppose there are two types of customers. One type gets to choose only one queue; each customer is of this type with probability $1-p$. The more privileged customer gets to choose two queues; each customer is of this type with probability p. The corresponding infinite system is governed by the following set of differential equations:

$$\frac{ds_i}{dt} = \lambda p(s_{i-1}^2 - s_i^2) + \lambda(1-p)(s_{i-1} - s_i) - (s_i - s_{i+1}).$$

The equilibrium point is given by $\pi_0 = \lambda$, $\pi_i = \lambda\pi_{i-1}(1-p+p\pi_{i-1})$. Note that this matches the supermarket model for $d = 1$ and $d = 2$ in the cases where $p = 0$ and $p = 1$, respectively. There does not appear to be a convenient closed form for the fixed point for other values of p.

This model has an interesting alternative interpretation. A customer who only has one choice is equivalent to a customer who has two choices, but erroneously goes to the wrong queue half of the time. Hence, the above system is equivalent to a two-choice system where customers make errors and go to the wrong queue with probability $\frac{1-p}{2}$. A model of this sort may therefore also be useful in the case where the information available from the chosen servers is unreliable.

4.4.2 Bounded buffers

In practice, we may have a system where the queue size has a maximum limit, say b. In this case, arriving customers that find queues filled are turned away. That is, for the supermarket model, if an arriving customer chooses d queues all of which have b customers already waiting, the customer leaves the system immediately without being served.

The state can be represented by a finite dimensional vector $(s_0, s_1, \ldots, s_b)$. Besides the expected time in the system, it would be useful to know the probability that a customer is turned away. The long-term probability that a customer is turned away can be determined by finding the fixed point; the probability a customer is turned away is then π_b^d. Alternatively, if one is interested in the number of lost customers over a certain interval of time from a specific starting state, one can add a variable to the state to count the number of customers turned away over time. The infinite system is given by the following equations:

$$\begin{aligned} \frac{ds_i}{dt} &= \lambda(s_{i-1}^d - s_i^d) - (s_i - s_{i+1}), \quad i < b; \\ \frac{ds_b}{dt} &= \lambda(s_{b-1}^d - s_b^d) - s_b. \end{aligned}$$

Note that at the fixed point for this problem, we do not have $\pi_1 = \lambda$. The total arrival rate of customers into the queues at the fixed point is $\lambda(1-\pi_b^d)$, as rejected customers do not enter the system. Since at the fixed point the total rate at which customers arrive must equal the rate at which they leave, we have $\pi_1 = \lambda(1 - \pi_b^d)$. Using the differential equations, we can develop a recurrence for the values of the fixed point π_i. This recurrence yields a polynomial equation for π_b, which can be shown to have a unique root between 0 and 1. Solving for π_b then allows us to compute the fixed point numerically. (These steps are shown in more detail in the explanation of the weak threshold model in Section 4.4.4.)

4.4.3 Closed models

In the closed model, at each time step exactly one non-empty queue, chosen uniformly at random, completes service, and the customer is immediately recycled back into the system by again choosing the shortest of d random queues. Let the number of customers that cycle through the system be αn. Note that the average number of customers per queue is α; this corresponds to the invariant $\sum_{i=1}^{\infty} s_i = \alpha$.

The infinite system is again very similar to that of the supermarket model. An important difference is that at each step, the probability that a customer leaves a server with i customers is $\frac{s_i - s_{i+1}}{s_1}$, since a random queue with at least one customer loses a customer. The corresponding differential equations are thus

$$\frac{ds_i}{dt} = s_{i-1}^d - s_i^d - \frac{s_i - s_{i+1}}{s_1}. \tag{4.7}$$

To find the fixed point, assume $\pi_1 = \beta$. Then inductively, we can solve to find $\pi_i = \beta^{\frac{d^i - 1}{d-1}}$; the correct value of β can be found by using the constraint $\sum_{i=1}^{\infty} \pi_i = \sum_{i=1}^{\infty} \beta^{\frac{d^i-1}{d-1}} = \alpha$. It is not surprising that the fixed point for the closed model looks similar to the fixed point for the supermarket model, as the closed model is exactly like the supermarket model, except that its state space is truncated to states where there are αn customers. (For further reference, see the State Truncation Property described in [62].)

A slightly different closed model was successfully analyzed (in a completely different manner) by Azar *et al.* in [11]. Their model describes the following scenario: suppose that each server represents a hash bucket, rather than a queue of customers, and at each step a random item is deleted from the hash table and a new item is inserted. We call this the *hashing closed model.* The corresponding infinite system is remarkably similar to the previous example (4.7):

$$\frac{ds_i}{dt} = s_{i-1}^d - s_i^d - i\frac{s_i - s_{i+1}}{\alpha}.$$

The fixed point for this system, however, does not appear to have a convenient closed form.

4.4.4 The threshold model

In some systems, limiting the amount of communication may be an important consideration. A threshold system reduces the necessary communication by only allowing a customer a second random choice if its first choice exceeds a fixed threshold. The customer begins by choosing a single queue uniformly at random: if the queue length at this first choice (excluding the incoming customer) is at most T, the customer lines up at that queue; otherwise, the customer chooses a second queue uniformly at random (with replacement). Two variations are now possible. In the *weak threshold model*, the customer waits at the second queue, regardless of whether it is longer or shorter than the first. In the *strong threshold model*, if both choices are over the threshold, the customer queues at the shorter of its two choices. (See Figure 4.2.) One could also expand both models so that a customer could possibly have several successive choices, with a different threshold set for each choice, up to any fixed number of choices; here we model only the case where a customer has at most two choices. Threshold systems have been shown to perform well both in theoretical models and in practice [29, 48, 77], although our results (such as the connection to density dependent Markov chains) appear to be new.

We first consider the weak threshold model. The rate at which a queue changes size is clearly dependent on whether a queue has more or less than T customers. We first calculate $\frac{ds_i}{dt}$ in the case $i \leq T+1$. Let $p_i = s_i - s_{i+1}$ be the fraction of queues with exactly i customers. An arriving customer becomes the ith customer in a queue if one of two events happen: either its first choice has $i-1$ customers, or its first choice has $T+1$ or more customers and its second choice has $i-1$ customers. Hence over a time interval dt the

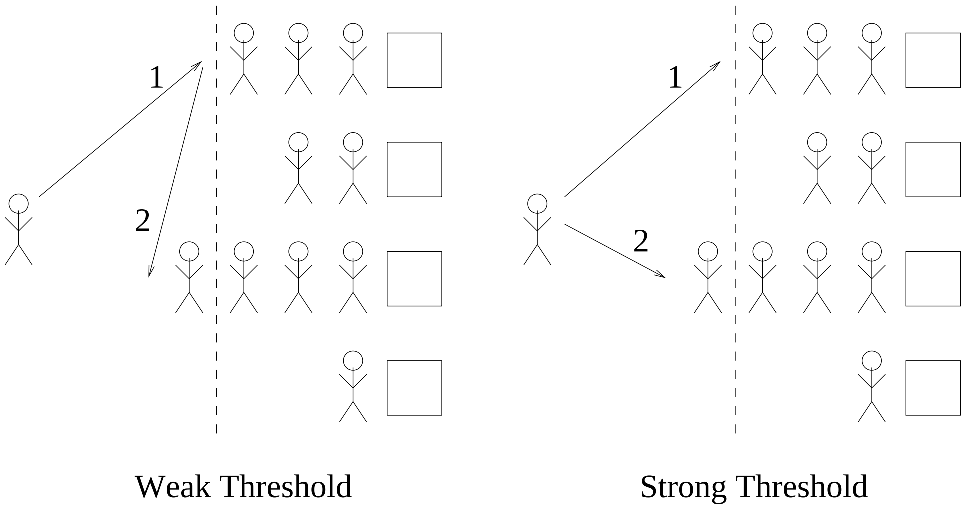

Figure 4.2: Weak and strong threshold models. A customer rechooses if and only if they would start behind the dashed line. In the weak model, the customer jumps to a second server, and may end up at one with a longer line (2). In the strong model, the customer goes to the shorter of the two lines (1).

expected number of jumps from queues of size $i-1$ to i is $\lambda n(p_{i-1} + s_{T+1}p_{i-1})$. Similarly, the expected number of jumps from queues of size i to $i-1$ is $np_i dt$. Hence we find

$$\begin{aligned}\frac{ds_i}{dt} &= \lambda(p_{i-1} + s_{T+1}p_{i-1}) - p_i\,, \quad i \leq T+1, \text{ or} \\ \frac{ds_i}{dt} &= \lambda(s_{i-1} - s_i)(1 + s_{T+1}) - (s_i - s_{i+1})\,, \quad i \leq T+1. \end{aligned} \tag{4.8}$$

The case where $i \geq T+1$ can be calculated similarly, yielding

$$\frac{ds_i}{dt} = \lambda(s_{i-1} - s_i)s_{T+1} - (s_i - s_{i+1})\,, \quad i > T+1. \tag{4.9}$$

We now seek to determine the fixed point. As usual, $\pi_0 = 1$ and, because at the fixed point the rate at which customers arrive must equal the rate at which they leave, $\pi_1 = \lambda$. In this case we also need to find the value of π_{T+1} to be able to calculate further values of π_i. Using the fact that $\frac{ds_i}{dt} = 0$ at the fixed point yields that for $2 \leq i \leq T+1$,

$$\pi_i = \pi_{i-1} - \lambda(\pi_{i-2} - \pi_{i-1})(1 + \pi_{T+1}). \tag{4.10}$$

Recursively plugging in, we find

$$\pi_{T+1} = 1 - \frac{(1-\lambda)[((1+\pi_{T+1})\lambda)^{T+1} - 1]}{(1+\pi_{T+1})\lambda - 1}.$$

Given the threshold T, π_{T+1} can be computed effectively by finding the unique root between 0 and 1 of the above equation. (The root is unique as the left hand side is increasing in π_{T+1}, while the right hand side is decreasing in π_{T+1}.) Note that in this system the π_i *do not* decrease doubly exponentially, although they can decrease very quickly if π_{T+1} is sufficiently small.

The strong threshold model is given by the following differential equations:

$$\frac{ds_i}{dt} = \lambda(s_{i-1} - s_i)(1 + s_{T+1}) - (s_i - s_{i+1}), \quad i \leq T+1; \tag{4.11}$$

$$\frac{ds_i}{dt} = \lambda(s_{i-1}^2 - s_i^2) - (s_i - s_{i+1}), \quad i > T+1. \tag{4.12}$$

For small thresholds, the behavior of this system is very similar to that of the supermarket system, as has been noted empirically previously in [29] and [77]. In fact, we next show that the strong threshold model is double exponentially decreasing, as one would expect from the differential equations (4.12).

Lemma 4.8 *The fixed point for the strong threshold model decreases doubly exponentially.*

Proof: To show that the fixed point decreases doubly exponentially, we note that it is sufficient to show that $\pi_{T+j+1} = \lambda\pi_{T+j}^2$ for all $j \geq 1$, from which the lemma follows by a simple induction. Moreover, to prove that $\pi_{T+j+1} = \lambda\pi_{T+j}^2$ for all $j \geq 1$, it is sufficient to show that $\pi_{T+2} = \lambda\pi_{T+1}^2$. That this is sufficient follows from equation (4.12) and the fact that $\frac{ds_i}{dt} = 0$ at the fixed point, from which we obtain

$$\lambda\pi_{i-1}^2 - \pi_i = \lambda\pi_i^2 - \pi_{i+1}$$

for $i \geq T+2$.

Hence, to prove the lemma, we now need only show that $\pi_{T+2} = \lambda\pi_{T+1}^2$. From equation (4.11) we have

$$\pi_{T+2} = \pi_{T+1} - \lambda(\pi_T - \pi_{T+1})(1 + \pi_{T+1}),$$

which can be written in the form

$$\pi_{T+2} - \lambda\pi_{T+1}^2 = (1+\lambda)\pi_{T+1} - \lambda(1 + \pi_{T+1})\pi_T. \tag{4.13}$$

We show that the right hand side of equation (4.13) is 0.

The recurrence (4.10), which also describes the fixed point for the strong threshold model for $2 \leq i \leq T+1$, yields that

$$\lambda(\pi_{i-2} - \pi_{i-1})(1 + \pi_{T+1}) = \pi_{i-1} - \pi_i.$$

Summing the left and right hand sides of the above equation for all values of i in the range $2 \leq i \leq T+1$ yields

$$\lambda(1 - \pi_T)(1 + \pi_{T+1}) = \lambda - \pi_{T+1},$$

or more conveniently,

$$\lambda(1 + \pi_{T+1})\pi_T = (1 + \lambda)\pi_{T+1}.$$

Hence the right hand side of equation (4.13) is 0 and the lemma is proved. ■

4.4.5 Load stealing models

Another paradigm for load distribution in a multiprocessor network is *load stealing.* In this discipline, jobs are allocated randomly to a group of processors, and underutilized processors seek out work from processors with higher utilization. Often this approach is more communication-efficient than standard load balancing strategies, since if all processors are busy, they do not need to communicate with each other in an attempt to distribute the load.

A basic load stealing scheme, which has been shown to perform well in static problems [20], is to have a processor that has completed all its tasks choose another processor at random and steal a job from that processor if possible. If no job is found, the processor attempts to steal again from another random processor, and so on until a job is found. We call the processor attempting to steal a *thief,* and say that it is looking for a *victim.*

Our framework allows us to consider dynamic variants of this problem. We begin with a simple model: tasks are generated at each processor as a Poisson process of rate $\lambda < 1$. Tasks require an exponentially distributed amount of service time before completing, with a mean of 1. The task times are not known to the processors. Tasks are served by a First In First Out (FIFO) policy. We shall assume that stealing can be done instantaneously, so that the stolen task joins the queue of the thief immediately. Tasks will be stolen from the end of the victim's queue; the victim must have at least two jobs for a job to be stolen. We also simplify the system by allowing a processor to make only one steal attempt when it empties; after that it waits for a new arrival (although the model is easily extended).

When a processor that completes its final job attempts to find a victim, the probability of success is just s_2, the probability of choosing a victim processor whose queue has at least two jobs. Hence we must reduce the departure rate by a factor of $1 - s_2$, yielding

$$\frac{ds_1}{dt} = \lambda(s_0 - s_1) - (s_1 - s_2)(1 - s_2). \tag{4.14}$$

For $i > 1$, s_i decreases whenever a processor with load i completes a job, or when a job is stolen. The rate at which thieves attempt to steal jobs is just $(s_1 - s_2)$, the rate at which processors complete their final job, and hence we find

$$\frac{ds_i}{dt} = \lambda(s_{i-1} - s_i) - (s_i - s_{i+1}) - (s_i - s_{i+1})(s_1 - s_2)\ ,\ i \geq 2. \tag{4.15}$$

The fixed point is easily found, since $\pi_0 = 1$ and $\pi_1 = \lambda$. Using equation (4.14) to solve for π_2 yields

$$\pi_2 = \frac{1 + \lambda - \sqrt{1 + 2\lambda - 3\lambda^2}}{2},$$

and from equation (4.15) we have by induction that, for $i \geq 2$,

$$\pi_i = \pi_2 \left(\frac{\lambda}{1 + \lambda - \pi_2} \right)^{i-2}.$$

Hence, for $i \geq 2$, we have that π_i decreases geometrically, albeit at a faster rate than if there were no stealing, in which case the fixed point is $\pi_i = \lambda^i$. An interpretation of this phenomenon is that, once a queue hits a certain load, it is as though the service rate has increased due to the stealing. Alternatively, if we think of the service rate as always being scaled to unity, then it is as though the arrival rate falls from λ to $\frac{\lambda}{1+\lambda-\pi_2}$ when we introduce stealing.

This approach can be generalized in several directions, including varying the time to transfer a task and the load at which processors seek to steal tasks; the work of Eager *et al.* examines some of these possibilities [30]. It may also be used to model other work stealing strategies, such as that proposed by Rudolph, Slivkin-Allalouf, and Upfal in [68].

4.4.6 The edge orientation problem

We now examine a rather different type of load balancing problem. Consider the complete graph on n vertices. At each time step an edge, chosen independently from past choices and uniformly at random, arrives and must be oriented toward one of its adjacent vertices. The *weight* of a vertex $w(v)$ is the difference between its indegree and the outdegree. The goal is to minimize the weight discrepancy, that is, to minimize the function $\max_v |w(v)|$. The algorithm we consider orients each arriving edge towards the vertex of smaller weight (with ties broken arbitrarily). This model is called the *edge orientation problem* in [8], where it was shown that the maximum weight discrepancy is $O(\log\log n)$ in equilibrium with high probability. This problem arises in the analysis of the carpool problem, in which people attempt to determine a fair protocol for dividing the task of driving over several days. The edge orientation problem corresponds to a carpool problem where every day two random people must choose a driver between them, and they make their choice based on the difference between the number of times each has driven and been driven.

The process can be described as a Markov chain. Following our previous examples, here we embed the chain in an infinite dimensional space where s_i is the fraction of vertices of weight at least i. In this case we do not have $s_0 = 1$, and i can be negative. Also, we will let p_i be the fraction of vertices with weight exactly i.

To write the differential equation describing this process, consider first the probability that an incoming edge increases the number of vertices of weight at least i. This can happen in one of two ways: first, both endpoints of the edge could have weight $i-1$, in which case the new edge will leave one with weight i. The probability of this event is $p_{i-1}^2 - O(\frac{1}{n})$. In the limiting case the $O(\frac{1}{n})$ term goes to 0 and so we shall drop it[3]; a more formal justification can be found in [32] or [53]. Alternatively, one of the endpoints could have weight $i-1$ and the other could have weight at least $i+1$. (If the endpoints have weight $i-1$ and i, then there is no overall change in the system!) The probability that this event happens is $2s_{i+1}p_{i-1}$. Similarly, one can determine the probability that an arrival causes a vertex to drop from weight i to weight $i-1$; the resulting differential equation is

$$\frac{ds_i}{dt} = p_{i-1}^2 + 2s_{i+1}p_{i-1} - (p_i^2 + 2p_i(1 - s_{i-1})).$$

Using the fact that $p_i = s_i - s_{i+1}$ we can simplify to:

$$\frac{ds_i}{dt} = s_{i-1}^2 - s_{i+1}^2 - 2(s_i - s_{i+1}).$$

[3]We have avoided this problem in other models by having customers choose servers with replacement. If customers choose without replacement, we again have this problem; however, it should be clear that this will not change the fixed point for the corresponding infinite systems.

We determine the fixed point under the assumption that we begin with an empty system; that is, one with no edges initially. In this case, from the underlying symmetry, we must have $p_i = p_{-i}$ for all time in the infinite system. Hence, for all time,

$$s_0 + s_1 = \sum_{i=0}^{\infty} p_i + \sum_{i=1}^{\infty} p_i = \sum_{i=0}^{\infty} p_i + \sum_{i=-\infty}^{-1} p_i = \sum_{i=-\infty}^{\infty} p_1 = 1.$$

Also, since at the fixed point $\frac{ds_i}{dt} = 0$, we have

$$\sum_{i=1}^{\infty} \frac{ds_i}{dt} = \sum_{i=1}^{\infty} \left[s_{i-1}^2 - s_{i+1}^2 - 2(s_i - s_{i+1}) \right] = s_0^2 + s_1^2 - 2s_1 = 0$$

at the fixed point. Solving the system of two equations and two unknowns yields

$$\pi_0 = \frac{1}{\sqrt{2}},\ \pi_1 = 1 - \frac{1}{\sqrt{2}},$$

and all other values of π_i can be derived from these initial two using the fact that $\frac{ds_i}{dt} = 0$ at the fixed point. Moreover, by examining the summation $\sum_{i=k+1}^{\infty} \frac{ds_i}{dt}$, we can derive that $\pi_k^2 + \pi_{k+1}^2 - 2\pi_{k+1} = 0$, from which it is easy to derive that the fixed point decreases doubly exponentially.

4.5 Simulations

In this section, we illustrate the effectiveness of the infinite system approach by examining a few of the models in more detail. We compare the predictions obtained from the infinite system model with simulations for the weak threshold model of Section 4.4.4 and the customer type model of Section 4.4.1. As an interesting byproduct of these comparisons, we demonstrate that these schemes, which use even less coordination than the supermarket model, distribute the load remarkably well. For both models, results are based on the average of ten simulations of 100,000 time steps; since the simulations started with no customers in the system, the first 10,000 time steps were ignored.

4.5.1 The weak threshold model

We consider the weak threshold scheme of Section 4.4.4 (where customers who make a second choice always queue at their second choice) with 100 queues at various arrival rates in Table 4.3. For arrival rates up to 95% of the service rate, the predictions are within approximately 2% of the simulation results; with smaller arrival rates, the prediction is even more accurate.

The approximation breaks down as the arrival rate nears the service rate. At 99% of the service rate, deviations of around 10% are noted with 100 queues; however, for systems with 500 queues the predictions are within 2% even at this arrival rate, as shown in Table 4.4. This is similar to the behavior of the supermarket model, as shown previously in Table 3.1 and Table 3.2.

As one might expect, threshold schemes do not perform as well as the supermarket model (Tables 3.1 and 3.2). It is worth noting, however, that the weak threshold scheme performs almost as well, which is somewhat surprising in light of the difference in the behavior of the tails (exponential versus doubly exponential dropoff). In many applications threshold schemes may be suitable, or even preferable, because in most instances only a single message and response are necessary. In the strong threshold model, performance is even better, and the difference between the threshold strategy and always choosing the shorter of two queues can be quite small.

4.5.2 Customer types

Recall that in the customer type model of Section 4.4.1, the parameter p measures the proportion of customers that choose from two queues instead of just one. We first examine the expected time in the infinite system model as p varies. For small fixed λ, one finds that as p increases the expected time falls almost linearly. At higher arrival rates, the drop becomes highly non-linear, with a sharp fall for small p and a more gradual drop as p approaches 1 (see Figure 4.3).

There is a good informal intuition for why this is the case. Consider the case when p is close to zero. If we temporarily ignore the customers that choose two queues, we have a system of independent queues each with arrival rate $\lambda(1-p)$. When λ is close to one, even a small value of p can make a substantial difference to the expected delay in this system. Adding back in the customers that choose two queues then has a comparatively small effect. Similarly, when p is close to one, then suppose we temporarily ignore the customers that only choose one queue. This system is not too different from a system where all customers choose two queues; adding back in the other customers only has a small effect. Even a small advantage in choosing a shorter queue can therefore lead to dramatic improvements,

λ	Threshold	Simulation	Prediction	Relative Error (%)
0.50	0	1.3360	1.3333	0.2025
	1	1.4457	1.4444	0.0900
	2	1.6323	1.6313	0.0613
	3	1.7695	1.7694	0.0057
0.70	0	1.9635	1.9608	0.1377
	1	1.8144	1.8074	0.3873
	2	2.0150	2.0109	0.2039
	3	2.2601	2.2570	0.1374
0.80	0	2.7868	2.7778	0.3240
	1	2.2493	2.2346	0.6578
	2	2.3518	2.3387	0.5601
	3	2.6192	2.6122	0.2680
0.90	0	5.2943	5.2535	0.7766
	1	3.5322	3.4931	1.1194
	2	3.1497	3.1067	1.3841
	3	3.2903	3.2580	0.9914
	4	3.6098	3.5839	0.7227
0.95	1	6.1120	5.9804	2.2005
	2	4.5767	4.4464	2.9305
	3	4.2434	4.1274	2.8105
	4	4.3929	4.3061	2.0158
	5	4.7426	4.6722	1.5068
	6	5.1640	5.1065	1.1260
0.99	4	8.1969	7.4323	10.2875
	5	7.5253	6.8674	9.5800
	6	7.6375	6.9369	10.0996
	7	7.8636	7.2925	7.8313
	8	8.3157	7.7823	6.8540
	9	8.8190	8.3385	5.7624

Table 4.3: Simulations versus estimates for the weak threshold model: 100 queues.

λ	Threshold	Simulation	Prediction	Relative Error (%)
0.99	4	7.6561	7.4323	3.0112
	5	7.0286	6.8674	2.3473
	6	7.0623	6.9369	1.8077
	7	7.3896	7.2925	1.3315
	8	7.8685	7.7823	1.1076
	9	8.4153	8.3385	0.9210

Table 4.4: Simulations versus estimates for the weak threshold model: 500 queues.

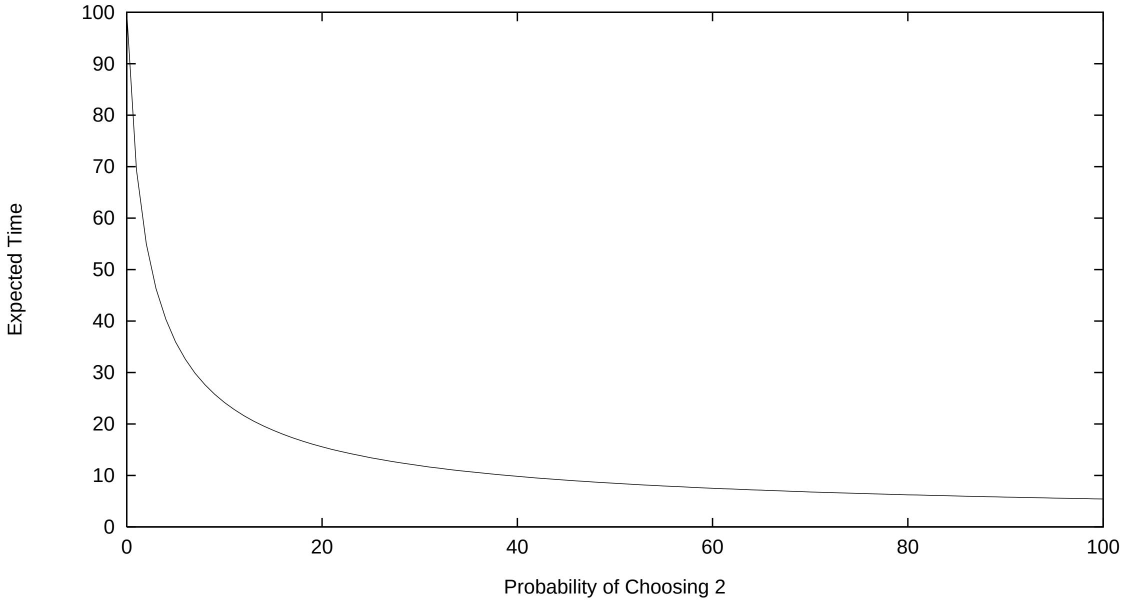

Figure 4.3: Expected time versus probability (p) of choosing two locations ($\lambda = 0.99$).

while a small probability of error for a customer does not drastically hurt performance, so the strategy is robust under errors. Our experience with the infinite system is borne out by simulations, as shown in Table 4.5.

4.6 Convergence and stability of infinite systems

In Chapter 3, we demonstrated that trajectories in the supermarket model converge to the fixed point exponentially making use of a somewhat complex potential function. We would like to prove similar results for the various dynamic systems we have described in this chapter.[4] For several of these systems, the proof of Theorem 3.6 can be modified in a straightforward fashion.

In some cases, however, proving convergence is difficult. For several systems, instead of proving convergence, it is much easier to prove the weaker property of *stability* of the fixed point. For our purposes, we may adopt the following definition of stability:

[4]It is worth recalling that for static systems, convergence to a fixed point is not an issue. For example, in Section 4.2, where we studied the static system of Azar *et al.*, we considered the system over a fixed time interval, and hence the question of convergence to the fixed point did not apply.

λ	p	Simulation	Prediction	Relative Error (%)
0.5	0.1	1.8765	1.8767	0.0107
	0.5	1.5266	1.5257	0.0590
	0.9	1.3094	1.3076	0.1377
0.7	0.1	2.9407	2.9425	0.0612
	0.5	2.0914	2.0884	0.1437
	0.9	1.6898	1.6843	0.3265
0.8	0.1	4.1315	4.1307	0.0194
	0.5	2.6351	2.6283	0.2587
	0.9	2.0526	2.0426	0.4896
0.9	0.1	7.0212	7.0208	0.0057
	0.5	3.7490	3.7247	0.6524
	0.9	2.7899	2.7601	1.0797
0.95	0.1	11.1007	11.0823	0.1660
	0.5	5.0845	5.0075	1.5377
	0.9	3.6698	3.5662	2.9050
0.99	0.1	24.5333	24.1874	1.4301
	0.5	8.9365	8.4716	5.4878
	0.9	6.2677	5.7582	8.8483

Table 4.5: Simulations versus estimates for two customer types: 100 queues.

Definition 4.9 *A fixed point $\vec{\pi}$ for a system $d\vec{s}/dt = f(\vec{s})$ is* stable[5] *if, for any $\epsilon > 0$, there exists a $\delta = \delta(\epsilon)$ such that $|\vec{s}(t) - \vec{\pi}| < \epsilon$ for all $t \geq 0$ whenever $|\vec{s}(0) - \vec{\pi}| < \delta$.*

A fixed point is hence stable if, once the trajectory is within ϵ of the fixed point, it will never again go beyond $\delta(\epsilon)$ of it; that is, once it becomes close to the fixed point, it stays close. To prove stability, we will actually show a stronger property for several systems; namely, that the L_1-distance to the fixed point is non-increasing everywhere. Then the fixed point is clearly stable from Definition 4.9, with $\delta(\epsilon) = \epsilon$.

We note here that there are other notions of convergence besides exponential convergence (Definition 3.5) and stability, but we will not require them here. A good introduction to dynamical systems in this context is the book by Miller and Michel [58]. We also suggest other works by these authors [56, 57], which provide a deeper treatment of the subject.

Rather than prove results for every specific model, we first prove some general theorems on stability and exponential convergence that apply to several of the models we

[5] We emphasize that stability of the *fixed point* is a different notion from the stability of the *system*, as described in Lemma 3.1.

have suggested. We believe these results are interesting in their own right and will be useful in the future for studying other systems. We then explain how the results apply to the systems we have studied.

4.6.1 General stability and convergence results

In this section, we consider general systems governed by the equations $\frac{ds_i}{dt} = f_i(\vec{s})$ for $i \geq 1$, with fixed point $\vec{\pi} = (\pi_i)$. Let $\epsilon_i(t) = s_i(t) - \pi_i$, with the understanding that for $i < 1$ or i larger than the dimension of the state space we fix $\epsilon_i = 0$. We shall drop the explicit dependence on t when the meaning is clear. For convenience, we shall consider only systems where $s_i(t) \in [0,1]$ for all t, and hence $\epsilon_i(t) \in [-\pi_i, 1-\pi_i]$ for all t. This restriction simplifies the statements of our theorems and can easily be removed; however, all the systems described in this chapter meet this condition.

We examine the L_1-distance $D(t) = \sum_{i \geq 1} |\epsilon_i(t)|$. In the case where our state space is countably infinite dimensional, the upper limit of the summation is infinity, and otherwise it is the dimension of the state space. As in Theorem 3.6, $\frac{dD}{dt}$ will denote the right-hand derivative. We shall prove that $\frac{dD}{dt} \leq 0$ everywhere; this implies that $D(t)$ is non-increasing over time, and hence the fixed point is stable.

For many of the systems we have examined, the functions f_i have a convenient form: they can be written as polynomial functions of the s_j with no product terms $s_j s_k$ for $j \neq k$. This allows us to group together terms in dD/dt containing only ϵ_i, and consider them separately. By telescoping the terms of the derivative appropriately, we can show the system is stable by showing that the sum of the terms containing ϵ_i are at most 0.

Theorem 4.10 *Suppose we are given a system $d\epsilon_i/dt = \sum_j g_{i,j}(\epsilon_j)$, where the $g_{i,j}$ satisfy the following conditions:*

1. $g_{i,i}(x) = -\sum_{j \neq i} g_{j,i}(x)$ *for* $x \in [-\pi_i, 1-\pi_i]$;
2. *for all* $i \neq j$, $\mathrm{sgn}(g_{j,i}(x)) = \mathrm{sgn}(x)$ *for* $x \in [-\pi_i, 1-\pi_i]$.

Then for $D(t) = \sum_{i=1}^{\infty} |\epsilon_i(t)|$ we have $dD/dt \leq 0$, and hence the fixed point is stable.

As a sample application of the above theorem, recall from Theorem 3.6 that for the supermarket model $d\epsilon_i/dt = -[\lambda(2\pi_i\epsilon_i + \epsilon_i^2) + \epsilon_i] + \lambda(2\pi_i\epsilon_{i-1} + \epsilon_{i-1}^2) + \epsilon_{i+1}$. Hence, in the notation of Theorem 4.10, for the supermarket system $g_{i,i}(x) = -[\lambda(2\pi_i x + x^2) + x]$, $g_{i,i+1}(x) = x$,

$g_{i,i-1}(x) = \lambda(2\pi_i x + x^2)$, and all other $g_{i,j}$ are 0. It is simple to check that these $g_{i,j}$ satisfy the conditions of the theorem, and therefore the supermarket system is stable (as we already knew).

Proof: We group the ϵ_i terms in dD/dt, and show that the sum of all terms involving ϵ_i is at most 0. We examine the case where $\epsilon_i > 0$; the case where $\epsilon_i < 0$ is similar, and the case where $\epsilon_i = 0$ is trivial.

The terms in ϵ_i sum to $h(\epsilon_i) = g_{i,i}(\epsilon_i)\,\mathrm{sgn}^*(\epsilon_i) + \sum_{j\neq i} g_{j,i}(\epsilon_i)\,\mathrm{sgn}^*(\epsilon_j)$, where $\mathrm{sgn}^*(\epsilon_j)$ is 1 if $\epsilon_j > 0$ or $\epsilon_j = 0$ and is nondecreasing, and $\mathrm{sgn}^*(\epsilon_j)$ is -1 if $\epsilon_j < 0$ or $\epsilon_j = 0$ and is decreasing. Note that, from the conditions of the theorem, $g_{i,i}(\epsilon_i)\,\mathrm{sgn}^*(\epsilon_i) \leq 0$, as $g_{i,i}(\epsilon_i) \leq 0$ when $\epsilon_i > 0$ and $g_{i,i}(\epsilon_i) \geq 0$ when $\epsilon_i < 0$. Since $g_{i,i}(x) = -\sum_{j\neq i} g_{j,i}(x)$, we may conclude that regardless of the values of $\mathrm{sgn}^*(\epsilon_j)$, the value of $h(\epsilon_i)$ is at most 0. Hence $dD/dt \leq 0$, and this suffices to show that the fixed point is stable. ■

A simple generalization of Theorem 4.10 allows us to prove convergence, using a weighted form of the potential function as in Theorem 3.6.

Theorem 4.11 *Suppose we are given a system* $d\epsilon_i/dt = \sum g_{i,j}(\epsilon_j)$, *and suppose also that there exists an increasing sequence of real numbers* w_i *(with* $w_0 = 0$*) and a positive constant* δ *such that the* w_i *and* $g_{i,j}$ *satisfy the following conditions:*

1. $\sum_j w_j g_{j,i}(x) \leq -\delta w_i |x|$ *for* $x \in [-\pi_i, 1-\pi_i]$;
2. *for all* $i \neq j$, $\mathrm{sgn}(g_{j,i}(x)) = \mathrm{sgn}(x)$ *for* $x \in [-\pi_i, 1-\pi_i]$.

Then for $\Phi(t) = \sum_{i=1}^{\infty} w_i |\epsilon_i(t)|$, *we have that* $d\Phi/dt \leq -\delta\Phi$, *and hence from any initial point where* $\sum_i w_i|\epsilon_i| < \infty$ *the process converges exponentially to the fixed point in* L_1*-distance.*

Proof: We group the ϵ_i terms in $d\Phi/dt$ as in Theorem 4.10 to show that the sum of all terms involving ϵ_i is at most $-\delta w_i|\epsilon_i|$. We may conclude that $d\Phi/dt \leq -\delta\Phi$ and hence $\Phi(t)$ converges exponentially to 0. Also, note that we may assume without loss of generality that $w_1 = 1$, since we may scale the w_i. Hence we may take $\Phi(t)$ to be larger than the L_1-distance to the fixed point $D(t)$, and thus the process converges exponentially to the fixed point in L_1-distance. ■

Proving convergence thus reduces to showing that a suitable sequence of weights w_i satisfying Condition 1 of Theorem 4.11 exist, which, as in the case of the supermarket model, is quite often straightforward.

Theorems 4.10 and 4.11 apply directly to several of the models we have mentioned. The following corollary is almost immediate:

Corollary 4.12 *The infinite systems for the following models have stable fixed points: gamma distributed service times with aware customers (Section 4.3), customer types (Section 4.4.1), bounded buffers (Section 4.4.2), and closed hashing (Section 4.4.3).* ∎

It is only slightly more difficult to show that these models converge exponentially to their fixed points; we omit the straightforward proofs.

Corollary 4.13 *The infinite systems for the following models converge exponentially to their corresponding fixed points: gamma distributed service times with aware customers, customer types, bounded buffers, and closed hashing.* ∎

For the other systems we have studied, Theorems 4.10 and 4.11 do not immediately apply. Even in cases where the function $d\epsilon_i/dt$ may not have the exact form required for Theorems 4.10 or 4.11, however, the technique of examining the terms in each ϵ_i separately still often proves effective. For example, the same technique allows one to prove that the fixed point for the closed model given by the equations (4.7) in Section 4.4.3 is stable. Also, it can be used to show that the fixed point for the edge orientation problem in Section 4.4.6 is stable, and that further the infinite system for this problem converges exponentially to the fixed point under the assumption that the system begins empty.

Showing stability or convergence for the remaining systems (gamma distributed service times with unaware customers, the threshold models, and the load stealing model) proves much more difficult. We examine some of these difficulties in the next section, where we will focus on the threshold models. We believe, however, that these problems and most other simple systems may be solvable using a variant of these techniques.

4.6.2 The threshold model: stability and convergence

To demonstrate how we can use the idea of Theorem 4.10 even if the derivatives are not of the exact form in the statement of the theorem, we first show that the weak threshold model is stable. It is convenient to write the derivatives $d\epsilon_i/dt$ obtained from equations (4.8) and (4.9) in the following form:

$$\frac{d\epsilon_i}{dt} = \lambda(\epsilon_{i-1} - \epsilon_i)(1 + \pi_{T+1}) - (\epsilon_i - \epsilon_{i+1}) + \lambda\epsilon_{T+1}(s_{i-1} - s_i)\,, \quad i \leq T+1\,; \tag{4.16}$$

$$\frac{d\epsilon_i}{dt} = \lambda(\epsilon_{i-1} - \epsilon_i)\pi_{T+1} - (\epsilon_i - \epsilon_{i+1}) + \lambda\epsilon_{T+1}(s_{i-1} - s_i)\,, \quad i > T+1. \tag{4.17}$$

Notice that we have made all the terms appear linear in ϵ_i by leaving terms of the form $\lambda\epsilon_{T+1}(s_{i-1} - s_i)$ unexpanded.

Theorem 4.14 *The fixed point of the weak threshold model is stable.*

Proof: We examine the potential function given by the L_1-distance $D(t) = \sum_{i=1}^{\infty} |\epsilon_i(t)|$, and show that $\frac{dD}{dt} \leq 0$. As in Theorem 4.10 we collect all terms with a factor of ϵ_i. For $i \neq T+1$, it is simple to verify that all terms are linear in ϵ_i, and that the coefficient of sum of all such terms is at most 0. For example, for $i < T+1$, the sum of the terms in ϵ_i is

$$(-\lambda(1 + \pi_{T+1}) - 1)\epsilon_i \operatorname{sgn}^*(\epsilon_i) + \lambda(1 + \pi_{T+1})\epsilon_i \operatorname{sgn}^*(\epsilon_{i+1}) + \epsilon_i \operatorname{sgn}^*(\epsilon_{i-1}),$$

which is at most 0. The case $i > T+1$ is similarly straightforward.

The only difficulty arises in the ϵ_{T+1} term. Note the different form of the first expression on the right hand side of (4.16) and (4.17) : one has a factor of π_{T+1}, and one has a factor of $1 + \pi_{T+1}$. Hence, in gathering the terms in ϵ_{T+1}, we have the following sum:

$$(-\lambda(1 + \pi_{T+1}) - 1)\epsilon_{T+1} \operatorname{sgn}^*(\epsilon_{T+1}) + \lambda\pi_{T+1}\epsilon_{T+1} \operatorname{sgn}^*(\epsilon_{T+2})$$
$$+\epsilon_{T+1} \operatorname{sgn}^*(\epsilon_T) + \epsilon_{T+1} \sum_{j=1}^{\infty} \lambda(s_{j-1} - s_j) \operatorname{sgn}^*(\epsilon_j).$$

Let us suppose that ϵ_T, ϵ_{T+1}, and ϵ_{T+2} are all strictly positive; all other cases are similar. Then the above summation reduces to

$$-\lambda\epsilon_{T+1} + \epsilon_{T+1} \sum_{j=1}^{\infty} \lambda(s_{j-1} - s_j) \operatorname{sgn}^*(\epsilon_j).$$

The largest value the second expression can take is when $\operatorname{sgn}^*(\epsilon_j) = 1$ for all j, in which case it is $\lambda\epsilon_{T+1}$. Hence, regardless of the signs of the remaining ϵ_i, we find that the coefficient of the sum of the terms in ϵ_{T+1} is also at most 0. ■

A similar proof shows that the load stealing system given by equations (4.14) and (4.15) is stable.

We now explain why our approach to show exponential convergence seems to fail for the weak threshold model. To prove exponential convergence, we would seek an increasing set of weights w_i for the potential function $\Phi(t) = \sum_{i=1}^{\infty} w_i |\epsilon_i(t)|$ so that $d\Phi/dt \leq -\delta\Phi$. The problem, as one might expect, comes from the terms in ϵ_{T+1}. Suppose that $\epsilon_i > 0$ for all i, and collect all terms in ϵ_{T+1}. The discrepancy in telescoping the derivatives $\frac{d\epsilon_{T+1}}{dt}$ and $\frac{d\epsilon_{T+2}}{dt}$ yields an additional term $-\lambda w_{T+1}\epsilon_{T+1}$. If the w_i sequence is increasing, this term does not seem to balance with the sum of the rightmost terms in equations (4.16) and (4.17), which sum to $\epsilon_{T+1} \sum_i \lambda w_i (s_{i-1} - s_i)$. This is in contrast to Theorem 4.14, where in the case where all w_i are 1, the term $-\lambda\epsilon_{T+1}$ matches the summation $\epsilon_{T+1} \sum_i \lambda(s_{i-1} - s_i)$, so that we may conclude that the terms in ϵ_{T+1} sum to at most 0.

It is perhaps not surprising that this problem arises. If all coordinates s_i besides s_{T+1} had reached π_i, the system would not smoothly continue toward its fixed point: $\frac{d\epsilon_i}{dt}$ would be non-zero for every i. Apparently the system's strong dependence on s_{T+1} makes it more challenging to prove exponential convergence. For the strong threshold model, however, these problems do not arise, as the value of s_{T+1} does not directly affect all other coordinates. Hence, we can show that the strong threshold model does converge exponentially.

As in Theorem 4.14, it will help us to rewrite the derivatives $\frac{d\epsilon_i}{dt}$ for the infinite system of the strong threshold model obtained from the equations (4.11) and (4.12) in the following form:

$$\frac{d\epsilon_i}{dt} = \lambda(\epsilon_{i-1} - \epsilon_i)(1 + \pi_{T+1}) - (\epsilon_i - \epsilon_{i+1}) + \lambda\epsilon_{T+1}(s_{i-1} - s_i), \quad i \leq T+1; \quad (4.18)$$

$$\frac{d\epsilon_i}{dt} = \lambda(\epsilon_{i-1}^2 + 2\pi_{i-1}\epsilon_{i-1} - \epsilon_i^2 - 2\pi_i\epsilon_i) - (\epsilon_i - \epsilon_{i+1}), \quad i > T+1. \quad (4.19)$$

Theorem 4.15 *The strong threshold model converges exponentially to its fixed point.*

Proof: We shall find an increasing sequence w_i and $\delta > 0$ such that for $\Phi(t) = \sum_i w_i |\epsilon_i(t)|$, we have $d\Phi/dt = -\delta\Phi$. As in Theorem 3.6 and Theorem 4.11, the proof will depend on finding a sequence w_i such that the terms of $d\Phi/dt$ in ϵ_i sum to at most $-\delta w_i|\epsilon_i|$. In fact, any sequence satisfying

$$w_{i+1} \leq w_i + \frac{w_i(1-\delta) - w_{i-1}}{\lambda(1+\pi_{T+1})}, \quad i < T+1 \quad (4.20)$$

$$w_{i+1} \leq w_i + \frac{w_i(1-\delta) - w_{i-1}}{\lambda(1+2\pi_i)}, \quad i \geq T+1 \tag{4.21}$$

will suffice, and it is easy to verify that such sequences exist, as in Theorem 3.6. That this condition suffices can be easily checked in the same manner as Theorem 3.6 for all ϵ_i except ϵ_{T+1}. The difficulty lies in the extraneous $\lambda\epsilon_{T+1}(s_{i-1} - s_i)$ terms in equation (4.18).

To conclude the theorem, we bound the sum of the terms in ϵ_{T+1}. We consider here only the case where all ϵ_i are positive; other cases are similar. The sum of all the terms in ϵ_{T+1} is

$$(-\lambda(1+\pi_{T+1}) - 1)w_{T+1}\epsilon_{T+1}\,\mathrm{sgn}^*(\epsilon_{T+1}) + \lambda(2\pi_{T+1} + \epsilon_{T+1})w_{T+2}\epsilon_{T+1}\,\mathrm{sgn}^*(\epsilon_{T+2}) +$$
$$w_T\epsilon_{T+1}\,\mathrm{sgn}^*(\epsilon_T) + \epsilon_{T+1}\sum_{j=1}^{T+1} w_j\lambda(s_{j-1} - s_j)\,\mathrm{sgn}^*(\epsilon_j).$$

If all ϵ_i are positive this reduces to

$$(-\lambda(1+\pi_{T+1})-1)w_{T+1}\epsilon_{T+1}+\lambda(2\pi_{T+1}+\epsilon_{T+1})w_{T+2}\epsilon_{T+1}+w_T\epsilon_{T+1}+\epsilon_{T+1}\sum_{j=1}^{T+1} w_j\lambda(s_{j-1}-s_j).$$

As the w_i are increasing, the term $\epsilon_{T+1}\sum_{j=1}^{T+1} w_j\lambda(s_{j-1} - s_j)$ can be bounded above by

$$\epsilon_{T+1}\sum_{j=1}^{T+1} w_{T+1}\lambda(s_{j-1} - s_j) = \epsilon_{T+1}w_{T+1}\lambda(1 - \pi_{T+1} - \epsilon_{T+1}).$$

Hence the sum of the terms in ϵ_{T+1} is bounded above by

$$(-\lambda(2\pi_{T+1} + \epsilon_{T+1}) - 1)w_{T+1}\epsilon_{T+1} + \lambda(2\pi_{T+1} + \epsilon_{T+1})w_{T+2}\epsilon_{T+1} + w_T\epsilon_{T+1},$$

and it is easily checked that equation (4.21) is sufficient to guarantee that this sum is at most $-\delta w_{T+1}\epsilon_{T+1}$. ∎

In contrast with the weak threshold model, in the strong threshold model the effect of s_{T+1} is only felt in terms s_i with $i \leq T+1$, and this distinction is enough for us to prove exponential convergence.

4.7 Open problems and future work

In the past two chapters, we have explored how to analyze dynamic load balancing models using the infinite system approach. The primary remaining open question for the models we have studied is to prove exponential convergence for the weak threshold system and the other systems for which we have only been able to prove stability. Also, a comparison argument relating exponential service times and constant service times for the supermarket model would be interesting, as suggested in Section 4.2.

Another very interesting question, which seems to require different techniques, is to consider how the network topology affects these simple randomized load balancing schemes. For example, in the supermarket model, we might think of the servers as being arranged in a ring. An incoming customer will choose from two (or more) servers, but these choices must be *adjacent* in the ring. Simulations suggest that having two choices in this model also provides dramatic improvements over having just a single choice, but we do not yet have a suitable approach for analyzing this problem.

We believe that the infinite system approach may have application in other domains as well. For example, we are currently working with the Priority Encoding Transmission group at the International Computer Science Institute to see if the infinite system approach can be useful in the design of simple erasure codes. The idea of using infinite systems has also been applied to recent work in genetic algorithms, such as the work on quadratic dynamical systems by Rabani, Rabinovich, and Sinclair [63]. We hope that the work of this thesis will encourage this type of analysis in other areas.

We also believe that it is important to apply the ideas in this thesis, especially the idea of the power of two choices, to actual systems. Already some practitioners have noted the relevance of our ideas to their work. Bestavros suggests that the paradigm of the power of two choices holds for his "load profiling" algorithms for distributed real-time systems [18]; Dahlin notes the relationship between our results and his studies for load balancing in caches in networks of workstations [25]. Distributed systems, especially networks of workstations, appear to be growing larger and more common [17], suggesting that the applicability of the simple randomized load balancing schemes we have studied will grow in the future. We hope that, by providing a strong theoretical basis for the power of two choices, we can increase the impact of this idea on the development of real systems.

Bibliography

[1] R. Abraham, J. Marsden, and T. Raitu. *Manifolds, Tensor Analysis, and Applications.* Addison-Wesley, 1983.

[2] I. J. B. F. Adan. *A compensation approach for queueing problems.* CWI (Centrum voor Wiskunde en Informatica), 1994.

[3] I. J. B. F. Adan, G. van Houtum, and J. van der Wal. Upper and lower bounds for the waiting time in the symmetric shortest queue system. *Annals of Operations Research*, 48:197–217, 1994.

[4] I. J. B. F. Adan, J. Wessels, and W. H. M. Zijm. Analysis of the symmetric shortest queue problem. *Stochastic Models*, 6:691–713, 1990.

[5] I. J. B. F. Adan, J. Wessels, and W. H. M. Zijm. Analysis of the asymmetric shortest queue problem. *Queueing Systems*, 8:1–58, 1991.

[6] M. Adler, S. Chakrabarti, M. Mitzenmacher, and L. Rasmussen. Parallel randomized load balancing. In *Proceedings of the 27th ACM Symposium on the Theory of Computing*, pages 238–247, 1995.

[7] N. Alon, J. Spencer, and Paul Erdös. *The Probabilistic Method.* John Wiley and Sons, 1992.

[8] M. Atjai, J. Aspnes, M. Naor, Y. Rabani, L. Schulman, and O. Waarts. Fairness in scheduling. In *Proceedings of the Sixth Annual ACM/SIAM Symposium on Discrete Algorithms*, pages 477–485, 1995.

[9] Y. Azar, A. Broder, and A. Karlin. Online load balancing. In *Proceedings of the 33rd IEEE Symposium on Foundations of Computer Science*, pages 218–225, 1992.

[10] Y. Azar, A. Broder, A. Karlin, and E. Upfal. Balanced allocations. Journal version, preprint.

[11] Y. Azar, A. Broder, A. Karlin, and E. Upfal. Balanced allocations. In *Proceedings of the 26th ACM Symposium on the Theory of Computing*, pages 593–602, 1994.

[12] Y. Azar, B. Kalyanasundaram, S. Plotkin, K. Pruhs, and O. Waarts. Online load balancing of temporary tasks. In *WADS 1993*, Lecture Notes in Computer Science 709, pages 119–130, 1993.

[13] Y. Azar, J. Naor, and R. Rom. The competitiveness of online assignment. In *Proceedings of the 3rd ACM/SIAM Symposium on Discrete Algorithms*, pages 203–210, 1992.

[14] K. Azuma. Weighted sums of certain dependent random variables. *Tôhoku Mathematical Journal*, 19:357–367, 1967.

[15] A. D. Barbour. Networks of queues and the method of stages. *Advances in Applied Probability*, 8:584–591, 1976.

[16] A. D. Barbour, L. Holst, and S. Janson. *Poisson Approximation.* Oxford University Press, 1992.

[17] The Berkeley NOW Project. The Berkeley NOW Project: Project overview. Located at http://now.cs.berkeley.edu/index.html.

[18] A. Bestavros. Load profiling in distributed real-time systems. Preprint.

[19] W. H. Beyer, editor. *CRC Handbook of Mathematical Sciences: 6th Edition.* CRC Press, 1987.

[20] R. D. Blumofe and C. E. Leiserson. Scheduling multithreaded computations by work stealing. In *Proceedings of the 35th Annual IEEE Conference on Foundations of Computer Science*, pages 356–368, 1994.

[21] B. Bollobás. *Random Graphs.* Academic Press, London, 1985.

[22] A. Broder, A. Frieze, C. Lund, S. Phillips, and N. Reingold. Balanced allocations for tree-like inputs. *Information Processing Letters*, 55(6):329–332, 1995.

[23] A. Broder, A. Frieze, and E. Upfal. On the satisfiability and maximum satisfiability of random 3-CNF formulas. In *Proceedings of the 4th ACM-SIAM Symposium on Discrete Algorithms*, pages 322–330, 1993.

[24] A. Broder and A. Karlin. Multi-level adaptive hashing. In *Proceedings of the 1st ACM/SIAM Symposium on Discrete Algorithms*, 1990.

[25] M. Dahlin. *Serverless Network File Systems*. PhD thesis, University of California, Berkeley, 1995.

[26] K. Deimling. *Ordinary Differential Equations in Banach Spaces*. Springer-Verlag, 1977. Lecture Notes in Mathematics. Vol. 96.

[27] A. Dembo and O. Zeitouni. *Large Deviations Techniques and Applications*. Jones and Bartlett Publishers, 1993.

[28] M. Dietzfelbinger and F. Meyer auf der Heide. Simple, efficient shared memory simulations. In *Proceedings of the Fifth Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 110–119, 1993.

[29] D. L. Eager, E. D. Lazokwska, and J. Zahorjan. Adaptive load sharing in homogeneous distributed systems. *IEEE Transactions on Software Engineering*, 12:662–675, 1986.

[30] D. L. Eager, E. D. Lazokwska, and J. Zahorjan. A comparison of receiver-initiated and sender-initiated adaptive load sharing. *Performance Evaluation Review*, 16:53–68, March 1986.

[31] D. L. Eager, E. D. Lazokwska, and J. Zahorjan. The limited performance benefits of migrating active processes for load sharing. *Performance Evaluation Review*, 16:63–72, May 1988. Special Issue on the 1988 SIGMETRICS Conference.

[32] S. N. Ethier and T. G. Kurtz. *Markov Processes: Characterization and Convergence*. John Wiley and Sons, 1986.

[33] M. R. Garey and D. S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

[34] L. A. Goldberg, Y. Matais, and S. Rao. An optical simulation of shared memory. In *Proceedings of the Sixth Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 257–267, 1994.

[35] G. Gonnet. Expected length of the longest probe sequence in hash code searching. *Journal of the ACM*, 28(2):289–304, April 1981.

[36] L. Green. A queueing system with general-use and limited-use servers. *Operations Research*, 33:168–182, 1985.

[37] T. Hagerup and C. Rüb. A guided tour of Chernoff bounds. *Information Processing Letters*, 33:305–308, February 1990.

[38] B. Hajek. Asymptotic analysis of an assignment problem arising in a distributed communications protocol. In *Proceedings of the 27th Conference on Decision and Control*, pages 1455–1459, 1988.

[39] M. Harchol-Balter and D. Wolfe. Bounding delays in packet-routing networks. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on the Theory of Computing*, pages 248–257, 1995.

[40] N. Johnson and S. Kotz. *Urn Models and Their Application.* John Wiley and Sons, 1977.

[41] A. Kamath, R. Motwani, K. Palem, and P. Spirakis. Tail bounds for occupancy and the satisfiability threshold conjecture. In *Proceedings of the 35th IEEE Symposium on Foundations of Computer Science*, 1994.

[42] R. M. Karp, M. Luby, and F. Meyer auf der Heide. Efficient PRAM simulation on a distributed memory machine. In *Proceedings of the 24th ACM Symposium on the Theory of Computing*, pages 318–326, 1992.

[43] R. M. Karp and M. Sipser. Maximum matchings in sparse random graphs. In *Proceedings of the 22nd IEEE Symposium on Foundations of Computer Science*, pages 364–375, 1981.

[44] R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the 27th Conference on Decision and Control*, pages 352–358, 1990.

[45] F. P. Kelly. *Reversibility and Stochastic Networks.* John Wiley and Sons, 1979.

[46] L. Kleinrock. *Queueing Systems, Volume I: Theory.* John Wiley and Sons, 1976.

[47] V. F. Kolchin, B. A. Sevsat'yanov, and V. P. Chistyakov. *Random Allocations.* V. H. Winston & Sons, 1978.

[48] T. Kunz. The influence of different workload descriptions on a heuristic load balancing scheme. *IEEE Transactions on Software Engineering*, 17:725–730, 1991.

[49] T. G. Kurtz. Solutions of ordinary differential equations as limits of pure jump Markov processes. *Journal of Applied Probability*, 7:49–58, 1970.

[50] T. G. Kurtz. Limit theorems for sequences of jump Markov processes approximating ordinary differential processes. *Journal of Applied Probability*, 8:344–356, 1971.

[51] T. G. Kurtz. Strong approximation theorems for density dependent Markov chains. *Stochastic Processes and Applications*, 6:223–240, 1978.

[52] T. G. Kurtz. *Approximation of Population Processes. CBMS-NSF Regional Conf. Series in Applied Math.* SIAM, 1981.

[53] H. J. Kushner. *Approximation and Weak Convergence Methods for Random Processes, with Applications to Stochastic Systems Theory.* MIT Press, 1984.

[54] M. Luby and A. Wigderson. Pairwise independence and derandomization. Technical Report TR-95-35, International Computer Science Institute, 1995.

[55] P. D. MacKenzie, C. G. Plaxton, and R. Rajaraman. On contention resolution protocols and associated probabilistic phenomena. Department of Computer Science TR-94-06, University of Texas at Austin, April 1994. An extended abstract appears in STOC 1994.

[56] A. N. Michel and R. K. Miller. *Qualitative Analysis of Large Scale Dynamical Systems.* Academic Press, Inc., 1977.

[57] A. N. Michel and R. K. Miller. Stability theory for countably infinite systems of differential equations. *Tôhoku Mathematical Journal*, pages 155–168, 1980.

[58] R. K. Miller and A. N. Michel. *Ordinary Differential Equations.* Academic Press, Inc., 1982.

[59] M. Mitzenmacher. Bounds on the greedy routing algorithm for array networks. In *Proceedings of the Sixth Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 248–259, 1994. To appear in the *Journal of Computer Systems and Science.*

[60] M. Mitzenmacher. Constant time per edge is optimal on rooted tree networks. In *Proceedings of the Eighth Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 162–169, 1996.

[61] M. Mitzenmacher. Density dependent jump Markov processes and applications to load balancing. In *Proceedings of the 37th IEEE Symposium on Foundations of Computer Science*, 1996.

[62] R. D. Nelson. The mathematics of product form queuing networks. *ACM Computing Surveys*, 25(3):339–369, September 1992.

[63] Y. Rabani, Y. Rabinovich, and A. Sinclair. A computational view of population genetics. In *Proceedings of the 27th ACM Symposium on the Theory of Computing*, pages 83–92, 1995.

[64] R. Righter. and J. Shanthikumar. Extremal properties of the FIFO discipline in queueing networks. *Journal of Applied Probability*, 29:967–978, November 1992.

[65] S. M. Ross. Average delay in queues with non-stationary Poisson arrivals. *Journal of Applied Probability*, 15:602–609, 1978.

[66] S. M. Ross. *Introduction to Probability Models.* Academic Press, Inc., 1989.

[67] S.M. Ross. *Stochastic Models.* John Wiley and Sons, 1983.

[68] L. Rudolph, M. Slivkin-Allalouf, and E. Upfal. A simple load balancing scheme for task allocation in parallel machines. In *Proceedings of the Second Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 237–245, 1991.

[69] B. L. Schwartz. Queueing models with lane selection: A new class of problems. *Operations Research*, 22:331–339, 1974.

[70] A. Shwartz and A. Weiss. *Large Deviations for Performance Analysis.* Chapman & Hall, 1995.

[71] G. D. Stamoulis and J. N. Tsitsiklis. The efficiency of greedy routing in hypercubes and butterflies. *IEEE Transactions on Communications*, 42(11):3051–3061, November 1994. An early version appeared in the *Proceedings of the Second Annual ACM Symposium on Parallel Algorithms and Architectures*, p. 248-259, 1991.

[72] V. Stemann. *Contention Resolution Protocols in Hashing Based Shared Memory Simulations.* PhD thesis, University of Paderborn, 1995.

[73] V. Stemann. Parallel balanced allocations. In *Proceedings of the Eighth Annual ACM Symposium on Parallel Algorithms and Architectures*, 1996.

[74] R. R. Weber. On the optimal assignment of customers to parallel servers. *Journal of Applied Probability*, 15:406–413, 1978.

[75] W. Winston. Optimality of the shortest line discipline. *Journal of Applied Probability*, 14:181–189, 1977.

[76] R. Wolff. *Stochastic Modeling and the Theory of Queues.* Prentice-Hall, Inc., 1989.

[77] S. Zhou. A trace-driven simulation study of dynamic load balancing. *IEEE Transactions on Software Engineering*, 14:1327–1341, 1988.