Markov Chains, Computer Proofs, and Average-Case Analysis of Best Fit Bin Packing

E.G. Coffman, Jr.[1], D. S. Johnson[1], P. W. Shor[1], R. R. Weber[2]

Abstract. Many complex processes can be modeled by (countably) infinite, multi-dimensional Markov chains. Unfortunately, current theoretical techniques for analyzing infinite Markov chains are for the most part limited to three or fewer dimensions. In this paper we propose a computer-aided approach to the analysis of higher-dimensional domains, using several open problems about the average-case behavior of the Best Fit bin packing algorithm as case studies. We show how to use dynamic and linear programming to construct potential functions that, when applied to suitably modified multi-step versions of our original Markov chain, yield drifts that are bounded away from 0. This enables us to completely classify the expected behavior of Best Fit under discrete uniform distributions $U\{J,K\}$ when K is small. (Under $U\{J,K\}$, the allowed item sizes are i/K, $1 \le i \le J$, with all J possibilities equally likely.) In addition, we can answer yes to the long-standing open question of whether there exist distributions of this form for which Best Fit yields linearly-growing waste. The proof of the latter theorem relies on a 24-hour computation, and although its validity does not depend on the linear programming package we used, it does rely on the correctness of our dynamic programming code and of our computer's implementation of the IEEE floating point standard.

1. Introduction

There is growing interest in the relationship between online algorithms and Markov chains. For instance, Vitter and Krishnan [21] and Karlin et al. [11] have recently investigated the behavior of paging algorithms when the input request sequence is generated by a finite-state Markov process. Under such inputs, the paging algorithms themselves act like Markov processes, and can be analyzed as such. The analysis is simplified by the fact that these processes are themselves finite-state, since the paging algorithms use bounded memory. In many other algorithmic situations, however, the memory of the online algorithm is *not* bounded. In such cases, even if the input comes from the trivial one-state Markov chain (i.e., consists of a sequence of i.i.d. samples from some distribution) the algorithms may give rise to *infinite-state* Markov chains, and the analysis can become much more difficult, especially when the state space is multi-dimensional, as it typically will be.

In this paper we develop techniques for dealing with such chains, based on computer-generated and -analyzed potential functions. As a case study for the use of these techniques, we show how they can be used to resolve a longstanding open problem in the average-case analysis of bin packing algorithms.

In the one-dimensional bin packing problem, we are given a list L of N items of sizes $a_1, a_2, \cdots, a_N \in (0,1]$, and asked to pack the items into unit-capacity bins so as to minimize the number of bins used. This problem is NP-hard, so research has concentrated on algorithms whose goal is to find packings that are merely *close* to optimal. Of particular interest is the Best Fit algorithm, in which the items are packed online, with the first item going into the first bin, and each successive item going into a partially packed bin with the smallest gap large enough to hold the item. If no partially packed bin has room for the item, a new bin is started. Best Fit can be implemented to run in time $O(N\log N)$, and among online algorithms offers perhaps the best balance between worst and average-case packing performance. In particular, no online algorithm is known that beats Best Fit both in the worst-case and in the following standard average-case model.

Suppose L_N is a random list of length N, where the item sizes are taken independently from the $U(0,1]$ distribution, i.e., are uniformly distributed in the interval (0,1]. (This is by far the most-studied distribution in the bin packing literature, e.g., see [2,3,4,5,6,8,13,15,19,20].) For a given list L, let $\mathrm{OPT}(L)$ be the number of bins used in an optimum packing, and let $s(L)$ denote the sum of the item sizes. Note that for all lists L, $s(L) \le \mathrm{OPT}(L)$. Moreover, we have $\mathrm{E}[s(L_N)] = N/2$ and $\mathrm{E}[\mathrm{OPT}(L_N)] \sim N/2$ as $N \to \infty$ [13,15]. For a given algorithm A, let $A(L)$ denote the number of bins used when L is packed by A and define the *waste* $w_A(L)$ to be $A(L) - s(L)$.

[1] AT&T Bell Laboratories, Murray Hill, NJ 07974
[2] Cambridge University, Cambridge, England CB3 9ET

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.
25th ACM STOC '93-5/93/CA,USA
© 1993 ACM 0-89791-591-7/93/0005/0412...$1.50

It was shown in [19] that Best Fit has sublinear expected waste; in particular $E[w_{BF}(L_N)] = \Theta(N^{1/2}\log^{3/4}N)$. As a consequence, one can conclude that $\lim_{N\to\infty} E[BF(L_N)/OPT(L_N)] = 1$. All known online heuristics A with better worst-case performance than Best Fit (i.e., with smaller *asymptotic worst-case ratios* [10]) have $\lim_{N\to\infty} E[A(L_N)/OPT(L_N)] > 1$. Even the well-known *First Fit* algorithm (FF), which has the same asymptotic worst-case ratio (17/10) as Best Fit [10] and also has sublinear expected waste, has a faster growth rate for that waste: $E[w_{FF}(L_N)] = \Theta(N^{2/3})$ [4,19]. (In First Fit, each item is placed in the lowest-indexed bin that has room for it.) The only known online algorithm with better expected waste than Best Fit is the considerably more complicated algorithm of [20] that has expected waste $\Theta(N^{1/2}\log^{1/2}N)$, which is the best possible for *any* online algorithm; this algorithm however has an unbounded asymptotic worst-case ratio.

Thus Best Fit may well be the algorithm of choice in situations where robustness is required, and the question of just how robust it is becomes relevant. In particular, how well does it perform under distributions other than the standard $U(0,1]$ distribution discussed above? It is natural to begin by considering distributions $U(0,u]$ where item sizes are chosen uniformly from the interval $(0,u]$, $0 < u < 1$. Let $L_{N,u}$ be a random list of length N with items chosen according to $U(0,u]$. (Note that $L_N = L_{N,1}$). Random lists of this sort were studied in the context of the *Next Fit* online algorithm in [5,12], and in terms of off-line algorithms in [3], with very different conclusions.

For the off-line algorithm Best Fit Decreasing (BFD), in which the list is re-ordered by non-increasing item size before applying Best Fit, we have $E[w_{BFD}(L_{N,u})] = \Theta(N^{1/3})$ for $1/2 < u < 1$ and $E[w_{BFD}(L_{N,u})] = \Theta(1)$ for $u \le 1/2$ [3], both significant improvements over the behavior of BFD when $u = 1$, where $E[w_{BFD}(L_{N,1})] = \Theta(N^{1/2})$ [13,15]. For Next Fit (NF), the answers are qualitatively quite different. In Next Fit, which is a significantly less-effective algorithm than Best Fit, only the most recently started bin is available for packing, and if the current item doesn't fit, that bin is closed forever and the item starts a new bin. Next Fit has linear waste even for $u = 1$; it was shown in [5] that $\lim_{N\to\infty} E[NF(L_{N,1})/OPT(L_{N,1})] = 4/3$. Moreover, as u decreases, the asymptotic expected ratio for NF *increases*, hitting a maximum at around $u = .80$ before beginning to decline [12].

Is Best Fit more like its offline cousin BFD or its online relative NF? Experiments reported in [2,4] suggest that the latter is the case. Although as remarked above, Best Fit yields sublinear expected waste when $u = 1$, it appears to get drastically worse as soon as $u < 1$, yielding linear waste for all $u \in (0,1)$, with the constant of proportionality again peaking at around $u = 0.80$, while dropping to 0 and $u \to 0$. Unfortunately, although this phenomenon was first observed almost a decade ago [2], no one has yet proved the conjecture for any value of u. (The techniques used by Karmarkar in [12] for analyzing Next Fit do not apply, since they crucially depend on the fact that the packing decision is determined by a single parameter, the gap in the currently active bin.)

Recently, however, it was discovered that Best Fit has much the same behavior in a possibly more-tractable discrete version of the problem. In [4], the class of *discrete uniform distributions* $U\{J,K\}$ was introduced. Here the item sizes, instead of being chosen from a continuous real interval $(0,u]$, are chosen from a finite set of evenly spaced values within such an interval. In $U\{J,K\}$, the sizes allowed are of the form i/K, $1 \le i \le J$, all equally likely. Thus $U\{J,K\}$ can be viewed as a discretized version of $U(0,J/K]$. Although it was shown in [4] that many standard bin packing results do not carry over from the continuous to the discrete case, the phenomenon under discussion here does show up, at least in part.

For notational simplicity, we shall normalize our item sizes so that bins have capacity K and items are of size 1 through J. Let $L_{N,J,K}$ be a random N-item list with items chosen from the distribution $U\{J,K\}$. Table 1 presents experimentally determined average values of waste for $5 \le J \le 12$ and $J+2 \le K \le 14$. We ignore the cases where $K \in \{J,J+1\}$, since these correspond in the continuous case to $U(0,1]$, and it was already determined in [4] that they yield $\Theta(N^{1/2})$ expected waste. Values for $J \in \{3,4\}$ are also omitted, but are similar in flavor to those for $J = 5$. Here we report the results of only two runs for each distribution, one with N = 100,000,000 and one with N = 200,000,000, but for each we report the average value of the waste over all steps of the packing.

A plausible conclusion from this data is that the average waste is bounded by a constant for most pairs (J,K), but grows linearly for the pairs (8,11), (9,12), (9,13), (10,13), (10,14), and (11,14), all of which have values of J/K reasonably close to the value $u = 0.80$ for which $E[w_{BF}(L_{N,u})/N]$ appears to peak. Experiments with larger values of K suggest that as K increases, the set of J's that yield linear waste also grows larger, although small values of J continue to yield bounded waste, as does the value $J = K-2$. The following theorem from [4] confirms that waste remains bounded if J is sufficiently small with respect to K:

Theorem 1 [4]. *If* $J \le \sqrt{2K+2.25} - 1.5$, *then* $E[w_{BF}(L_{N,J,K})] = O(1)$.

It is also easy to see that expected waste is bounded when $J \le 2$ for all $K > J+1$. These, however, were until now the only known results covering the expected waste of Best Fit when $J < K-1$. Not only were there no results confirming the apparent examples of linear waste, but also none of the apparent examples of bounded expected waste

	K	7	8	9	10	11	12	13	14
J	5	2.28	0.94	0.74	0.62	0.60	0.56	0.53	0.52
		2.28	0.94	0.74	0.62	0.60	0.56	0.53	0.52
	6		2.45	1.70	0.90	0.80	0.68	0.66	0.60
			2.45	1.70	0.90	0.80	0.68	0.66	0.60
	7			4.58	1.97	1.55	1.00	0.84	0.75
				4.58	1.97	1.55	1.00	0.84	0.75
	8				4.72	5576.42	2.66	1.75	1.06
					4.71	10773.60	2.64	1.75	1.06
	9					7.97	13043.90	42642.23	6.41
						7.99	26696.40	84804.23	6.33
	10						8.06	57790.87	55961.10
							8.09	115493.92	112112.47
	11							12.50	51211.14
								12.46	102614.05
	12								12.64
									12.61

TABLE 1. Waste under Best Fit for $U\{J,K\}$, averaged over runs of 100- and 200-million items, as a function of J and K.

from Table 1 are covered by Theorem 1. Here is where Markov chains come in. A Markov chain M can be viewed as a pair (S,T), where S is the *state space* and $T{:}S\times S \to \mathbf{R}$ is a *transition function* satisfying $\Sigma_{s'\in S}T(s,s') = 1$ for all $s \in S$; if the current state is s, the probability that the next state is s' is $T(s,s')$.

A natural way to model an online bin packing algorithm by a Markov chain is to let the current packing be the current state. What makes Best Fit a promising candidate for analysis in this context is that the only details of the packing that need be considered are the numbers of bins with each possible gap, from 1 up to $K-1$. (We do not need to count bins with gaps of size 0, i.e., full bins, since we are only concerned with measuring waste.) If the best fit for an item is a bin of gap size i and there are many bins with gaps of that size, it doesn't matter which bin is chosen, at least insofar as the distribution of gap sizes in the resulting packing is concerned. Note that this is *not* the case for an algorithm such as First Fit, where the size of the gap into which an item is placed depends, not simply on the availability of bins with given gaps, but on the order in which the gaps appear in the packing.

Thus a Markov chain $M_{J,K}$ corresponding to a Best Fit packing process under input distribution $U\{J,K\}$ can have its states simply be $(K-1)$-tuples of nonnegative integers, where tuple $(s_1,s_2,\ldots,s_{K-1})$ represents a packing whose inventory of partially filled bins consists of precisely s_i bins with gap i, $1 \le i \le K-1$. As to transitions, note that if we are in state s and an item of size i arrives, the successor state, denoted by $s[i]$, is uniquely determined. For instance, suppose $K=4$ and $s=(3,1,0)$. If an item of size 1 arrives, the successor state is $(2,1,0)$; if an item of size 2 arrives it is $(3,0,0)$; if an item of size 3 arrives it is $(4,1,0)$; and if an item of size 4 arrives the state remains unchanged, since an empty bin becomes a full one, and neither is counted in our state. The transition function T can thus be defined by $T(s,s[i]) = 1/J$, $1 \le i \le J$ and $T(s,s') = 0$ if s' is not $s[i]$ for any i. For technical reasons having to do with Markov chain irreducibility, the actual state space of $M_{J,K}$ is restricted to those $(K-1)$-tuples that are reachable from the all-zero state by a sequence of transitions all having positive probability. (The all-zero state corresponds to a packing with no partially-filled bins, such as the initial empty packing.)

Using this model, one might hope to place bounds on the expected waste of Best Fit under $U\{J,K\}$ by deriving results about the chain $M_{J,K}$, in particular about its *stationary distribution*. Recall that such a distribution is an assignment of probabilities $p(s)$ to the states $s \in S$ such that $\Sigma_{s\in S}p(s) = 1$ and $\Sigma_{s'\in S}p(s')T(s',s) = p(s)$ for all $s \in S$. If one lets $\bar{p}$ be the vector of probabilities and views T as the corresponding matrix, this simply means $\bar{p}T = \bar{p}$.

There are two drawbacks to this approach. First, our state space is infinite and multidimensional. Although the theory of one-dimensional infinite Markov chains is well-developed, much is still unknown for higher dimensions. Limited results have been proved, but only for two and three dimensions [7,15,16,17]. For the open problem about $U\{8,11\}$ we must deal with a chain that is ostensibly of dimension 10. The second drawback is that chains like $M_{8,11}$, because they correspond to situations in which the expected waste is not bounded, are transient and do not have stationary distributions. We thus have our work cut out for us.

The remainder of this paper is organized as follows. In Section 2 we introduce the relevant terminology for dis-

	K	5	6	7	8	9	10	11	12	13	14
J	3	B-L2	B-L2	B-L2	B-Q1	B-Th	B-Th	B-Th	B-Th	B-Th	B-Th
	4		B-L4	B-L3	B-Q1	B-Q1	B-Q1	B-Q1	B-Q1	B-Q1	B-Th
	5			B-L23	B-Q1	B-Q1	B-Q1	B-Q1	B-Q1	B-Q1	B-Q1
	6				B-Q2	B-Q5	B-Q1	B-Q1	B-Q1	B-Q1	B-Q1
	7					B-Q7	B-Q15	B-Q2	B-Q2	B-Q1	B-Q1
	8						B-Q13	Ln-P	B-x	B-x	B-Q7
	9							B-x	Ln-P	Ln-x	B-x
	10								B-x	Ln-x	Ln-x
	11									B-x	Ln-x
	12										B-x

TABLE 2. Results proved for $E[w_{BF}(L_{N,J,K})]$.

cussing multidimensional chains, and summarize what is currently known about multidimensional Markov chains. These results are based on the use of a particular kind of potential function (called a *Lyapunov* function in the literature). Once one goes beyond three dimensions, there are no longer simple constructions for the needed functions. There are, however, algorithmic approaches to determining whether Lyapunov functions of certain standard types exist, and the remainder of Section 2 describes them. We show how linear programming can be used to test for the existence of both linear and quadratic Lyapunov functions, using an old lemma of Hajek [9] and a new one of our own. We also show how we can, in certain cases, circumvent the non-existence of the desired functions by switching attention to derived (*embedded*) Markov chains $M^t_{J,K}$, $t > 1$, which have the same state space as $M_{J,K} = (S_{J,K}, T_{J,K})$, but have product matrices $T^t_{J,K}$ for transition functions, i.e., chains in which one step corresponds to t steps of the original chain.

Section 3 then uses this technology to prove bounded expected waste for many of the entries in Table 1. A *proof* in this case starts with the construction by computer of a linear program that, although it has few variables, may sometimes have more than 100,000 constraints. The linear program is then solved using a standard LP package. The package we use is CPLEX™ (CPLEX is a trademark of CPLEX Optimization, Inc.), but the validity of our proofs is independent of the correctness of CPLEX, since after CPLEX generates a solution, we verify the validity of that solution using our own code. The correctness of our proofs does, however, depend on the correctness of our generation and checking programs (for which we shall present listings of key routines in the final paper), and on the fact that our computer runs properly and correctly implements the IEEE floating point standard [1].

In order to prove linear waste, which was our main goal, additional lemmas and techniques are required, and these are described in Section 4. We introduce a sequence of surrogate chains, involving the use both of "imaginary bins" and of transition functions where one step corresponds to a variable number of steps in the original chain. Our resulting proofs of linear waste for both $U\{8,11\}$ and $U\{9,12\}$ boil down to 24-hour computations of expected drifts for thousands of states over thousands of steps of the underlying chains. A summary of the results obtained in Sections 3 and 4 is given in Table 2. The full meaning of the entries will be explained later. For now it is enough to note that a prefix of *B* or *Ln* on an entry implies that our experiments suggested Bounded waste or Linear waste for the corresponding J,K pair. Note that this table has been expanded over Table 1 to include rows for $J \in \{3,4\}$. A *Th* after the hyphen means the result follows from Theorem 1. A *C*, *Q*, or *P* after the hyphen indicates that the result was proved by our new techniques. An *x* indicates that the problem remains open.

We conclude in Section 5 with a brief discussion of the limitations and further applicability of our techniques.

2. Multidimensional Markov Chains and Lyapunov Functions

Let $\mathbf{Z}_+$ represent the non-negative integers and let d be a positive integer. A *d-dimensional* Markov chain M is one whose state space S is a (possibly infinite) subset of $\mathbf{Z}^d_+$. Although the state space may be infinite, in cases of practical significance the transition function will typically have a finite description. Of particular interest is the class of chains with *limited state dependency* and *jump-boundedness*, as defined below.

Consider the derived incremental transition function $\theta_T : S \times \mathbf{Z}^d \to \mathbf{R}$, where for any $s \in S$ and $\delta \in \mathbf{Z}^d$ (negative components allowed), $\theta_T(s,\delta) = T(s, s+\delta)$ if $s+\delta \in S$ and $\theta_T(s,\delta) = 0$, otherwise. For any state $s = (s_1, s_2, \ldots, s_d) \in \mathbf{Z}^d_+$, let $m_k(s)$ be the vector obtained from s by replacing each component s_i by $min(s_i, k)$.

Definition. *A Markov chain $M = (S,T)$ has k-*limited *state* dependency *(is k-*limited *for short) if $\theta_T(s,\delta) = \theta_T(m_k(s),\delta)$ for all $s \in S$ and all $\delta \in \mathbf{Z}^d$.*

In other words, M is k-limited if the precise value of a state component s_i affects the incremental transition probabilities for that state only when the value is less than k; if the value is k or greater, the precise value doesn't matter. Note that the case $d = 1$, $k = 1$ includes the $M/M/1$ queue as well as other varieties of random walks in the presence of a barrier. Similarly, the case $d > 1$, $k = 1$ includes any Markovian network of queues (a basic model used in computer performance analysis), as well as more general random walks in the positive orthant of the d-dimensional lattice. (See [7] and its references.) In our bin packing application, it is easy to see that $M_{J,K}$ is 1-limited, since the gap into which an item is packed depends only on which gap counts are nonzero, not on how big the nonzero gap counts are. Similarly, our bin packing chains $M_{J,K}$ are *2-jump-bounded*, according to the following definition. For $x \in \mathbf{Z}^d$ let $\|x\| = \Sigma_{i=1}^d |x_i|$.

Definition. *A Markov chain $M = (S,T)$ is* jump-bounded *if $T(s,s') > 0$ implies $\|s-s'\| < j$ for some fixed constant j, in which case we say M is* j-jump-bounded.

Research to date on Markov chains of dimension $d > 1$ can be quickly summarized. Malyshev [16] considered the case of jump-bounded, 1-limited, 2-dimensional Markov chains and obtained simple criteria for ergodicity, null-recurrence, and transience based on T. Malyshev and Menshikov [17,18] extended this to a generalization of k-limited state dependency in which each component has its own individual bound k_i, and claimed similar results for $d = 3$ without proof. Fayolle [7] generalized the $d = 2$ case by further relaxing both the conditions of limited state dependency and jump boundedness.

In our bin packing application, however, we are interested in values of d significantly larger than 3 and in finer distinctions than that between ergodicity and transience. (A Markov chain is *ergodic* if its distribution converges to a unique stationary distribution; it is *transient* if no state has a positive probability of being visited infinitely often.) In particular, if we consider a random path $X_1, X_2, \ldots$ of our Markov chain starting with $X_1 = (0,0,\ldots,0)$, we are interested in the growth rate of $\mathrm{E}[\|X_n\|]$. (Note that for K fixed, the total waste in a packing lies within linear bounds of the number of partially filled bins.) Unfortunately, $\limsup_{n\to\infty} \mathrm{E}[\|X_n\|]$ can be either finite or infinite for an ergodic Markov chain, and although the limit is always infinite in the transient case, a variety of growth rates are possible.

Fortunately, general tools exist for analyzing particular Markov chains, even if no general theorem gives us our answers directly. In what follows we rely on the following specialization of a result of Hajek [9].

Lemma 2.1. *Suppose $X_1, X_2, \ldots$ is a random path of an irreducible Markov chain $M = (S,T)$, U is a finite subset of S, B and γ are positive reals, and ϕ is a function from S to $[0,\infty)$ such that*

(a) $\mathrm{Prob}(|\phi(X_{n+1}) - \phi(X_n)| > B) = 0$ *for all $n \geq 1$, and*

(b) $\mathrm{E}[\phi(X_{n+1}) - \phi(X_n) | X_n = s] \leq -\gamma$ *for all $n \geq 1$ and $s \in S - U$.*

Then $\limsup_{n\to\infty} \mathrm{E}[\phi(X_n)] < \infty$.

An *irreducible* Markov chain is one in which every state is reachable from every other by a path of positive probability. This property holds for our Best Fit chains because of our restriction that all states be reachable from the all-zero state. In the applied probability literature, a potential function ϕ satisfying (b) is called a *Lyapunov* function [7,14]. If M is aperiodic, as are our Best Fit Markov chains, then condition (b) by itself implies ergodicity, but ergodicity alone is not enough for our purposes. For this, we need the following immediate corollary.

Lemma 2.2. *Under the hypotheses of Lemma 2.1, if ϕ satisfies (a), (b), and*

(c) there exist constants $c,d \geq 0$ such that
$\|s\| \leq c\phi(s) + d$ *for all $s \in S$,*

then $\limsup_{n\to\infty} \mathrm{E}[\|X_n\|] < \infty$.

In searching for Lyapunov functions satisfying (c), a natural first class to consider is the set of linear functions $\phi(s) = \Sigma_{i=1}^d a_i s_i$ with $a_i > 0$, $1 \leq i \leq d$, since requiring that $a_i > 0$ for all i is perhaps the simplest way to insure that (c) holds. Note that condition (a) automatically holds for linear functions so long as the Markov chain is jump-bounded, as is the case with our Best Fit chains.

It is not difficult to see that the existence of a positive-coefficient linear Lyapunov function can be determined by linear programming when M is jump-bounded and has limited state dependency. Suppose M is k-limited. Then it is enough to show that (b) holds for the set S_k of $O(d^{k+1})$ states with no component exceeding k and at least one component having that value. All states with no component exceeding $k-1$ can be consigned to the set U of the lemma. For each state s and each i, $1 \leq i \leq d$, let $\Delta_i(s)$ be the expected change in s_i between X_n and X_{n+1} when $X_n = s$. Then the existence of the desired Lyapunov function can be shown to depend on the solution of the following linear program. (Note that the LP can only require that the a_i be nonnegative, not that they be positive, so the correspondence is not quite immediate.)

Maximize γ, subject to

$$\sum_{i=1}^{d} a_i \Delta_i(s) < -\gamma, \textit{ for all } s \in S_k, \tag{2.2}$$

$$a_i \geq 0,\ 1 \leq i \leq d, \textit{ and} \tag{2.3}$$

$$\sum_{i=1}^{d} a_i \leq 1. \tag{2.4}$$

This linear program is always feasible; the last constraint is added to insure that the program is bounded, but otherwise represents no restriction for our purposes.

Lemma 2.3. *If M is a k-limited, j-jump-bounded, d-dimensional Markov chain, then it has a positive-coefficient* *linear Lyapunov function if and only if the LP described by (2.1)-(2.3) has a solution with $\gamma > 0$.*

Proof. It is immediate that if $\gamma \le 0$, no linear Lyapunov function with all positive coefficients can exist. On the other hand, a solution $(\gamma, a_1, \ldots, a_d)$ with $\gamma > 0$ yields a Lyapunov function with the desired properties: the function $\phi(s) = \Sigma_{i=1}^d a_i s_i$ satisfies (b) for all states with a component of value k or bigger by (2.2) and the fact that M is k-limited. If all the a_i are positive we are done; otherwise obtain a new function by adding $\gamma/2dj$ to each a_i. The new function has all positive coefficients and will continue to satisfy (b) with γ replaced by $\gamma/2 > 0$. □

Unfortunately, for the situations in which we are interested, the solution to the linear program given by (2.1)-(2.3) all too often has $\gamma \le 0$, even for Markov chains where $\limsup_{n\to\infty} E[\|X_n\|] < \infty$. In this case we may want to expand the domain in which we look for Lyapunov functions. The obvious next step is to consider quadratic functions $\phi(s) = sQs^T$, where Q is a $d \times d$ matrix. We may assume without loss of generality that Q is symmetric, since if Q defines a function that satisfies (b), then so will $Q + Q^T$. Quadratic functions were used for $d = 2$ in [7], which suggested without much detail that the approach could be extended to arbitrary dimensions, and which again was not concerned with questions of boundedness and so ignored property (c). There are still straightforward ways to ensure this property, however. The simple approach we take here is to require that all diagonal components of Q be positive and all other components be non-negative. By a slightly abused analogy with the linear case, we shall call such a function a *positive-coefficient quadratic function.*

Satisfying property (b) is more of a problem. Note that even if M is k-limited, the expected value $E[\phi(X_{n+1}) - \phi(X_n) | X_n = s]$ is no longer independent of the precise values of the components s_i that are k or greater. If we let $\Delta(s) = (\Delta_1(s), \ldots, \Delta_d(s))$, we have

$$\phi(s+\Delta(s)) - \phi(s) = 2sQ\Delta(s)^T + \Delta(s)Q\Delta(s)^T \quad (2.4)$$

This holds because of our assumption that Q is symmetric. Fortunately, we can make the dependence on s work for us, given that the chains we are interested in are jump-bounded. If we can find a Q such that $sQ\Delta(s)^T$ goes to $-\infty$ as components of s currently greater than k get large, this term can be made to swamp any positive contribution from $\Delta(s)Q\Delta(s)^T$. All we need do is take the set U of Lemma 2.1 to contain all states with no component exceeding h, for a sufficiently large value of h.

We can thus solve for the components $q_{i,j}$, $1 \le i,j \le d$, of an appropriate Q using linear programming. Assume M is k-limited, and once again let S_k be the set of states whose maximum component equals k. If we let $n(s,k)$ be the number of components in state s that are equal to k, then our linear program will have $n(s,k)$ constraints for each state $s \in S_k$. The overall linear program will once again seek to maximize γ subject to

$$\text{for all } s \in S_k \text{ and } j \text{ such that } s_j = k, \quad (2.5)$$
$$\sum_{j=1}^{d} q_{i,j}\Delta_j(s) < -\gamma$$
$$q_{i,j} = q_{j,i},\ 1 \le i,j \le d, \quad (2.6)$$
$$q_{i,j} \ge 0,\ 1 \le i,j \le d, \quad (2.7)$$
$$\sum_{i=1}^{d} \Sigma_{j=1}^{d} q_{i,j} \le 1. \quad (2.8)$$

Lemma 2.4. *If M is a k-limited, j-jump-bounded, d-dimensional Markov chain, then it has a positive-coefficient quadratic Lyapunov function satisfying property (b) of Lemma 2.1 if and only if the linear program (2.5)-(2.8) has a solution with $\gamma > 0$.*

Proof. Relatively straightforward given the above discussion, and omitted because of space considerations, as will be all future proofs of lemmas. □

Thus the LP of (2.5)-(2.8) gives us a way of finding positive-coefficient quadratic Lyapunov functions satisfying (b) and (c). There is still one obstacle to applying Lemma 2.2, however: It is unlikely that such a quadratic Lyapunov function ϕ would obey condition (a), since for instance $(n+\Delta)^2 - n^2 = 2n\Delta + \Delta^2$ is not bounded independent of n. Moreover, as we can show by example, the fact that ϕ is a quadratic Lyapunov function for M does not in itself imply the conclusion of Lemma 2.1. (This and other examples will be detailed in the full paper.) Fortunately, we have the following result.

Lemma 2.5. *Suppose $\phi(s) = sQs^T$ is a positive-coefficient quadratic Lyapunov function satisfying (b) and (c) for a k-limited, j-jump-bounded, d-dimensional Markov chain M. Then $\phi' = \phi^{1/2}$ is a Lyapunov function for M satisfying (a), (b), and (c).*

Unfortunately, there remain many cases where $\limsup_{n\to\infty} E[\|X_n\|] < \infty$ and yet neither linear nor quadratic Lyapunov functions exist. How can this be? The problem is that certain classes of states must of necessity get worse before they can get better. This is where the idea comes in of considering "multistep" embedded Markov chains of the form $M^t = (S, T^t)$, where $T^t(s,s')$ is the probability that state s' is reached in t steps when s is the current state. Note that if M is k-limited and j-jump-bounded, then M^t is tk-limited and tj-jump-bounded, and so the above approaches to finding linear or quadratic Lyapunov functions still apply. Further note that the transition matrix T^t is simply the t'th power of the matrix T, and so is easily computed.

Computed is the operant word here, however. Although our techniques for finding Lyapunov functions involve solving linear programs and hence most likely will require some computation, one could still hope to end up with a relatively concise proof in the common case where $k = 1$ and d is relatively small. In the linear Lyapunov case, the linear programs would have at most $2^d - 1$ problem-specific constraints (one for each possible pattern of nonzeros in s), which is roughly 1000 for $d = 10$. For a proof, one would only need to write down the program and its solution (which could then be verified, albeit laboriously, by hand). In the quadratic Lyapunov case the program size can go up by a factor of d, but this would still be manageable with $d \le 7$, say.

Once we start considering k-limited chains with $k > 1$, however, as we need to for the t-step Markov chains, the numbers of constraints in the programs get out of hand for much smaller values of d. (The numbers are $\Theta((t+1)^d)$ in the linear case and $\Theta(d(t+1)^d)$ in the quadratic.) As we shall see, values of t as large as 19 were needed for some of our results. Moreover, even if we were to write down the linear program, its correctness would no longer be easy to verify by hand, since it is based on T^t, not T. Thus this proposed approach necessarily leads to proofs that only a computer can reasonably hope to verify. In Section 3 we shall talk about the computational details of applying Lemma 2.2 to prove bounded expected waste for Best Fit under distributions $U\{J,K\}$, and about how confident we can be in the correctness of those proofs.

Note also that Lemma 2.2 only applies when $\limsup_{n\to\infty} \mathrm{E}[\|X_n\|]$ is bounded. In those cases where we wish to prove that $\mathrm{E}[\|X_n\|]$ goes to infinity (as we do when trying to prove that Best Fit has linear waste under $U\{8,11\}$ and $U\{9,12\}$), more powerful techniques are required. In particular, as we shall see in Section 4, we'll need to find embedded chains and Lyapunov functions that yield appropriate drifts for *all* states, not just those outside some finite set.

3. Proving Expected Waste is Bounded

In applying the techniques outlined in the previous section to our Best Fit Markov chains, we first must look more carefully at the relevant state spaces. Since $M_{J,K}$ is 1-limited, the embedded multi-step chain $M^t_{J,K}$ is t-limited. Thus in our analysis we can restrict attention to those states in which no component exceeds t. We know *a priori* that there can be no more than $(t+1)^{K-1}$ such states, but the number of such states in the chains $M^t_{J,K}$ is actually substantially below this bound.

A major reason for this is our "technical requirement" that the state space of $M_{J,K}$ be by definition restricted to those states reachable by sequences with positive probability from the all-0 state, i.e., those vectors corresponding to packings that could actually be constructed starting with an empty packing. Under Best Fit a packing can never contain two bins whose total contents sum to K or less. (The later of the two bins could never have been started, as its first item would have fit into the gap in the earlier bin.) This means in particular that for states $(s_1,\ldots,s_{K-1})$ in $M^t_{J,K}$,

$$\textit{If } i \ge K/2 \textit{ and } s_i > 0 \textit{ then } s_j = 0 \quad \textit{for all } j \ne i \textit{ with } K-i \le j \le K-1. \tag{3.1}$$

For instance, there can be at most one $i \ge K/2$ such that $s_i > 0$. Let $n_{t,K}$ be the number of $(K-1)$-tuples with no component exceeding t that satisfy (3.1). The following is easy to verify.

Lemma 3.1.

(i) *If* $K = 2r+1$ *for some integer* $r \ge 0$, *then* $n_{t,K} = \Sigma_{i=0}^{r}(t+1)^i$.

(ii) *If* $K = 2r$ *for some integer* $r \ge 1$, *then* $n_{t,K} = 2(t+1)^{r-1} + \Sigma_{i=0}^{r-2}(t+1)^i$.

The actual number of states with no component exceeding t in $M^t_{J,K}$ may fall short of $n_{t,K}$, depending on the value of J. As a trivial example, if $J = 1$ then at most one component can be non-zero for *any* value of K. For programming simplicity, however, we shall ask that the Lyapunov functions we generate satisfy property (b) for all states counted by $n_{t,K}$, even those ruled out by considerations based on the value of J. This allows us to make use of a relatively straightforward invertible procedure for computing a mapping $g_{t,K}$ from the states to the integers $1,2,\ldots,n_{t,K}$. This mapping in turn provides us with a simple way for looping through all states, and as we shall see, yields significant reductions in space usage. We also use an invertible mapping $tern_K$ that maps $\{-1,0,+1\}^{K-1}$ to integers and allows us to encode incremental transitions.

Our programs for generating and for checking the relevant LP's have the same basic structure and most of the same code. (They are written in C and run on a single 25 Mhz MIPS™ processor inside a Silicon Graphics IRIS™ 4D/250 computer, where MIPS is a trademark of MIPS, Inc., and IRIS is a trademark of Silicon Graphics, Inc.) Most of the storage space is devoted to two 2-dimensional arrays. The first is `theta[]`, an $n_{t,K} \times J$ array of integers, that is used as a sparse representation for the incremental transition function corresponding to $T_{J,K}$. For each state s and item size j, `theta[`$g_{t,K}(s)$`,j]` $= tern_K(s[j]-s)$, where $s[j]$ is the state corresponding to the packing that would arise if an item of size j were added to the packing corresponding to s by the Best Fit rule. The entries in the `theta[]` array are precomputed once and for all at the beginning of the program. Note that we save considerable space here by encoding the increment vectors as integers.

The second array `delta[]` is an $n_{t,K} \times (K-1)$ array of double precision floating point numbers used to store and compute the values of $\Delta^t_i(s)$, i.e., the expected change in component i if one starts in state s and takes t steps of the Markov chain. Recall that these are the key coefficients in the LP's specified by (2.2)-(2.4) and (2.5)-(2.8). We compute the $\Delta^t_i(s)$'s by dynamic programming rather than by explicitly constructing the transition function $T^t_{J,K}$, since this is an easy way to take advantage of the sparsity of $T_{J,K}$. Moreover, for many of the t,K combinations we need to consider, the explicit computation of $T^t_{J,K}$ would be prohibitive, if only for space reasons. The numbers of states can be quite large, ranging up to $n_{t,K} = 254{,}906$ in our longest proof of bounded waste (for $J=8,K=14$). Moreover, when $t > 1$, $T^t_{J,K}$ can have a significantly higher proportion of nonzero entries than the $8n_{t,K}$ we get when $t = 1$.

The value of `delta[g(s),i]` is initially set to $\Delta_i^0(s) = 0.0$, the expected change in component i after $u = 0$ steps. In the uth stage of our dynamic programming process the values of $\Delta_i^u(s)$ for those states s with maximum component u are computed from the values $\Delta_i^{u-1}(s)$ for states with maximum component $u-1$. This yields the desired values (for states with maximum component equal to t) after t stages. Recalling that $m_v(s)$ is the vector obtained from s by replacing each component s_i of s by $\min(s_i,v)$, we base the computation at each stage on the following recurrence relation:

$$\Delta_i^u(s) = \frac{1}{J}\sum_{j=1}^{J}\left[s[j]_i - s_i + \Delta_i^{u-1}(m_{u-1}(s[j]))\right]$$

for $1 \le i \le K-1$ and all states $s \in S_{J,K}$ with maximum component equal to u. Once the relevant values of $\Delta_i^t(s)$ have been computed, it is a simple matter either to generate the LP, or to verify that a given solution actually does yield a positive value for γ.

Our main task in constructing the proofs of bounded waste summarized in Table 2 was finding values of t for which the desired linear or quadratic Lyapunov functions exist. A suffix *Lt* (*Qt*) in the table indicates that the proof was based on a positive-coefficient linear (quadratic) Lyapunov function for $M^t_{J,K}$, and, if $t > 1$, that no such linear (quadratic) Lyapunov function exists for $M^{t-1}_{J,K}$. In general, linear functions sufficed for $K \le 7$, and for $K \ge 8$ we switched to quadratic functions since they either worked with $t = 1$ or required a lower value of t than did linear functions.

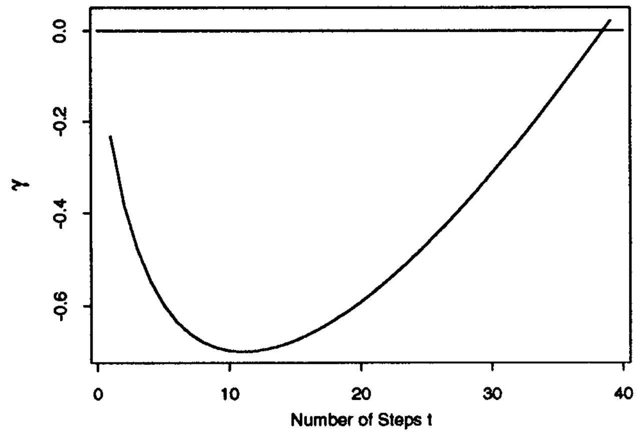

FIGURE 1. Solution values of linear Lyapunov function LP's for $M^t_{6,8}$, $1 \le t \le 39$.

For example, in the case of $U\{6,8\}$, we obtained a positive-coefficient quadratic Lyapunov function as soon as $t = 2$, whereas $t = 39$ was required for a linear one. The LP for the former required only about 4 kilobytes of storage, whereas that for the latter needed roughly 700. Figure 1 illustrates why t had to be so large in the linear case, and also how we measured progress in our attempts to find a suitable value of t. Here we chart the optimal values of γ in the linear-function LP's for $M^t_{6,8}$, $1 \le t \le 39$, with the bounding constraint (2.4) replaced by "$a_1 = 1$." (The replacement works equally well to keep the LP bounded, and for $t < 39$ prevents the uninformative solution of (2.2)-(2.4) that simply sets γ and all the a_i's equal to 0.) Note that as t increases γ first gets worse and then slowly increases until it finally becomes positive at $t = 39$. Such behavior was typical in those cases where a high value of t was required, both in the linear and quadratic case.

For those bounded space results we could not prove using our techniques (labeled *B-x* in Table 2), the bottleneck was the storage needed for LP's, rather than the time to generate them. The "hardest" bounded waste result we were able to prove was that for $U\{8,14\}$. For this distribution the lowest t for which a positive-coefficient quadratic Lyapunov function exists is $t = 7$. For this value it took only 37 minutes to generate the LP (and a comparable time for CPLEX to solve it and for us to check the solution). However, the LP itself required 56 megabytes on our disk, and generating and solving it used up roughly 100 megabytes of main memory. The disk space figure could have been reduced somewhat by choosing a more compact LP representation, but probably not by a factor of more than 2. For the case of $U\{8,13\}$, on which our techniques failed, the optimal γ's for the quadratic LP's were still getting worse by the time we got to $t = 8$ (analogous to the downward initial portion of the curve in Figure 1). Moreover, the LP for $t = 8$ already took up 44 megabytes of disk space and LP size was almost doubling with each increase of t by one. The outlook was similarly bleak for the other cases of bounded waste left open in Table 2.

How confident can we be in the proofs that we *have* managed to construct? Assuming our computer correctly implements our programs and that our programs correctly implement the techniques described above, the one remaining question has to do with the numerical precision of the computations. Here the key question is the accuracy of the computation that checks the solution to our LP to verify the negative drifts. This involves the operations needed to construct the LP as well as the operations needed to check it. The longest sequence of arithmetic operations involved in deriving any coefficient in our LP contains at most $t(J+1)$ operations, all on numbers no greater than t, where $t \le 20$ and $J \le 9$ in all our results. Since the IEEE Floating Point Standard [1] allows a relative error of at most 10^{-15} per double-precision operation, this means our final figures are all accurate to within 10^{-11}, assuming our computer correctly implements the standard, as claimed by the manufacturer. The checking phase adds a chain of up to $2d$ additional operations in the quadratic case, on numbers as big as 100, but the resulting cumulative relative error is still at most 10^{-10}. This is far too small to compromise our results, since the smallest positive γ we encountered was on the order of 10^{-3}.

4. Proving that Waste Grows Linearly

The proofs discussed in the previous section were relatively straightforward applications of the ideas described in Section 2. When it comes to proving that expected waste is not bounded but grows linearly, which was our main goal, additional ideas are needed. For these results we shall need the following new lemma (essentially a result about supermartingales) instead of Lemmas 2.1 and 2.2:

Lemma 4.1. *Suppose $X_1, X_2, \ldots$ is a random path from an irreducible Markov chain $M = (S,T)$, B and γ are positive reals, and ϕ is a function from S to $\mathbf{R}$ such that*

(a) $\mathrm{Prob}(|\phi(X_{n+1} - \phi(X_n)| > B) = 0$ *for all $n \geq 1$, and*

(b) $E[\phi(X_{n+1} - \phi(X_n) | X_n = s] \geq \gamma B$ *for all $n \geq 1$ and all $s \in S$.*

Then $\liminf_{n \to \infty} (\phi(X_n) - \gamma n + (1+\gamma')\sqrt{2n \log n}) > 0$ *for all $\gamma' > \gamma$ with probability 1.*

As indicated in Table 2 by the entries *Ln-P*, we have succeeded in proving that both $E[w_{BF}(L_{N,8,11})]$ and $E[w_{BF}(L_{N,9,12})]$ are $\Theta(N)$. From now on, we specialize to the case of $U\{8,11\}$. In order to apply Lemma 4.1, we will need to do more than simply go to multi-step versions of the chain $M_{8,11}$, as the effective state space will be too large. The chain to which we actually apply the lemma is derived in a two-step process. First we consider a Markov chain M^* that only approximates $M_{8,11}$, but is true to it in an asymptotic sense. Simulations suggest that under $U\{8,11\}$, essentially all the waste is contained in bins with gaps of size 1. Thus the count s_1 of bins with gaps of size 1 grows without limit and does not affect the packing process (except that it absorbs all items of size 1). In M^* we decouple the packing of 1-items from the packing of the other items. Any time a 1-item arrives, we simply decrement s_1, even if that means that s_1 should become negative. (Our state space is thus expanded to allow arbitrary negative integers as values for s_1, and only 0 is allowed as a value for s_{10}.) In this Markov chain, the expected state increment $\Delta^t(s)$ depends only on the values of s_2 through s_9, and only 4 of these values can exceed 1. We thus save a factor of about $(t+1)$ in the running time and space for the dynamic program that computes the values of $\Delta^t(s)$ over the time and space that would have been needed for $M_{8,11}$. It is necessary here to concentrate on linear potential functions; given Lemma 4.1 we can restrict attention to ones that have positive drift. In order to prove that M^* approximates $M_{8,11}$ we must also be sure that the drift is dominated by s_1. We thus restrict our search to linear functions $\phi(s) = \Sigma_{i=1}^{9} f_i s_i$ in which $f_1 = 1$ and $f_i \leq 0$, $2 \leq i \leq 9$.

Next, we do not simply go to a t-step version of M^* (for that we probably would have needed t substantially larger than 1000), but rather to a variable-step version M^{**}, based on two constants t_1 and t_2. A move of M^{**} is specified as follows: starting in state s, we perform steps of M^* until either we reach a state with an $s_i \geq t_1$ for some $i \geq 2$, or until we have gone t_2 steps, whichever comes first. In the former case we proceed for precisely t_1 additional steps; in the latter case we are already done. Thus a *step* of M^{**} corresponds to anywhere from t_1 to $t_1 + t_2$ steps of M^*. The value of t_1 is chosen as the minimum t for which there exists a linear function ϕ with significant positive drift on all states s with $\max\{s_i : i \geq 2\} \geq t$. (This t turns out to be 15, as was determined using the linear programming techniques developed in previous sections.) Using the ϕ corresponding to t_1, the value of t_2 is determined as the minimum t such that each state with $\max\{s_i : i \geq 2\} < t_1$ yields significant positive expected drift in the above process when $t_2 \geq t$. (With both t_1 and t_2 the threshold for "significant" positive drift was chosen so that rounding errors could not force the drift negative.) A value of $t_2 = 4818$ was determined by using dynamic programming to compute expected drifts. The procedure only had to consider states with $s_i \leq 15$ for $i \geq 2$, but the large number of steps forced the computation to take 24 hours on our machine; this computation was the heart of our proof and will be described in more detail in the full paper.

The results of the computation imply that Lemma 4.1 applies to M^{**}, and the lemma then implies that for M^{**} (and M^* as well), the value of s_1 goes to ∞ at a linear rate with probability 1. We can thus argue that there exists a state s of M^* with $s_1 > 0$ and an $\varepsilon > 0$ such that if M^* is in state s, the probability that the value of s_1 never again goes negative is at least ε. Consequently, the probability is at least ε that $M_{8,11}$ and M^*, both started in state s, will behave exactly the same (with s_1 going to ∞ at a linear rate). Since in $M_{8,11}$ any reachable state s can be reached with some probability $\varepsilon_s > 0$, this implies that in $M_{8,11}$ the value $\|X_n\|$ grows linearly with probability at least $\varepsilon\varepsilon_s > 0$. Consequently the expected value $E[\|X_n\|]$ must itself grow linearly, and hence $E[w_{BF}(L_{N,8,11})] = \Theta(N)$. Similar arguments work for $U\{9,12\}$ with $t_1 = 19$ and $t_2 = 814$, but again with 24 hours of computation because, although t_2 is smaller, the effective state space is larger.

5. Conclusion

The obvious open question is whether all the above machinery of Markov chains and computer proofs is necessary for the results we have obtained. Simple hand-checkable proofs would certainly be preferable. It is not clear, however, that the complexity of the current proofs can be avoided. For instance, in the case of $U\{8,11\}$, bins with gaps of size 1 are created at a rate only about 1% faster than the rate at which items of size 1 arrive (based on simulations of Best Fit on random lists), which does not leave much slack for the types of simplifying assumptions upon which intuitive proofs usually rely.

A second question is whether our techniques can be pushed further. Table 2 indicates the limits to which we can currently push them. All the still-open problems listed in the table appear to require solutions to LP's that are far too big for us to store, much less solve. Thus one must either wait for larger machines or look for more concise LP formulations, the latter being the more promising avenue. And it does not appear that we will be able to get these results for the discrete distributions $U\{J,K\}$ to provide more than suggestive insight into the apparently much more difficult case of the continuous distributions $U[0,u]$. In our opinion, the most promising direction for future work thus lies in applying our techniques to analyze Markov chains arising in new problem domains.

Acknowledgment. We thank David Applegate for invaluable assistance with the CPLEX package and for helpful comments on earlier drafts of this paper.

REFERENCES

1. *IEEE Standard for Binary Floating-Point Arithmetic*, ANSI/IEEE Std 754-1985, Institute for Electrical and Electronic Engineers, Inc., New York, 1985.
2. J. L. BENTLEY, D. S. JOHNSON, F. T. LEIGHTON, AND C. C. MCGEOCH, "An experimental study of bin packing," in *Proc. 21st Ann. Allerton Conf. on Communication, Control, and Computing*, University of Illinois, Urbana, IL, 1983, 51-60.
3. J. L. BENTLEY, D. S. JOHNSON, F. T. LEIGHTON, C. C. MCGEOCH, AND L. A. MCGEOCH, "Some unexpected expected behavior results for bin packing," in *Proceedings 16th Ann. ACM Symp. on Theory of Computing,* Association for Computing Machinery, New York, 1984, 279-288.
4. E. G. COFFMAN, JR, C. COURCOUBETIS, M. R. GAREY, D. S. JOHNSON, L. A. MCGEOCH, P. W. SHOR, R. R. WEBER, AND M. YANNAKAKIS, "Fundamental discrepancies between average-case analyses under discrete and continuous distributions: A bin packing case study," in *Proceedings 23rd Ann. ACM Symp. on Theory of Computing,* Association for Computing Machinery, New York, 1991, 230-240.
5. E. G. COFFMAN, JR, M. HOFRI, K. SO, AND A. C. YAO, "A stochastic model of bin packing," *Information and Control* **44** (1980), 105-115.
6. E. G. COFFMAN, JR AND G. S. LUEKER, *Probabilistic Analysis of Packing and Partitioning*, Wiley & Sons, New York, 1991.
7. G. FAYOLLE, "On random walks arising in queueing systems: Ergodicity and transience via quadratic forms as Lyapounov Functions – Part I," *Queueing Systems* **5** (1989), 167-184.
8. G. N. FREDERICKSON, "Probabilistic analysis for simple one- and two-dimensional bin packing algorithms," *Inform. Process. Lett.* **11** (1980), 156-161.
9. B. HAJEK, "Hitting-time and occupation-time bounds implied by drift analysis with applications," *Adv. Appl. Prob.* **14** (1982), 502-525.
10. D. S. JOHNSON, A. DEMERS, J. D. ULLMAN, M. R. GAREY, AND R. L. GRAHAM, "Worst case performance bounds for simple one-dimensional packing algorithms," *SIAM J. Comput.* **3** (1974), 299-325.
11. A. R. KARLIN, S. J. PHILLIPS, AND P. RAGHAVAN, "Markov paging," in *Proceedings 33rd Ann. Symp. on Foundations of Computer Science,'* IEEE Computer Society, Los Angeles, Calif., 1992, 208-217.
12. N. KARMARKAR, "Probabilistic analysis of some bin-packing algorithms," in *Proceedings 23rd Ann. Symp. on Foundations of Computer Science,'* IEEE Computer Society, Los Angeles, Calif., 1982, 107-111.
13. W. KNÖDEL, "A bin packing algorithm with complexity $O(n\log n)$ and performance 1 in the stochastic limit," in *Proc. 10th Symp. on Mathematical Foundations of Computer Science,* J. Gruska and M. Chytil (eds.), *Lecture Notes in Computer Science* **118**, Springer-Verlag, 1981, 369-378.
14. H. KUSHNER, *Introduction to Stochastic Control*, Holt, Rinehart and Winston, Inc., New York, 1971.
15. G. S. LUEKER, "An average-case analysis of bin packing with uniformly distributed item sizes," Report No. 181, Department of Information and Computer Science, University of California, Irvine, CA, 1982.
16. V. A. MALYSHEV, "Classification of two-dimensional positive random walks and almost-linear semi-martingales," *Soviet Math. Dokl.* **13** (1972), 136-139.
17. V. A. MALYSHEV AND M. V. MENSHIKOV, "Ergodicity, continuity and analyticity of countable Markov chains," *Trudi. Moscow Matem. Obshch* **39** (1979), in Russian. English version in *Trans. Moscow Math. Soc.*, **Issue 1** (1979), 1-48.
18. M. V. MENSHIKOV, "Ergodicity and transience conditions for random walks in the positive octant of space," *Soviet Math. Dokl.* **15** (1979), .
19. P. W. SHOR, "The average case analysis of some on-line algorithms for bin packing," *Combinatorica* **6** (1986), 179-200.
20. P. W. SHOR, "How to do better than best fit: Tight bounds for average-case on-line bin packing," in *Proceedings 32nd Ann. Symp. on Foundations of Computer Science,'* IEEE Computer Society, Los Angeles, Calif., 1990, 752-759.
21. J. S. VITTER AND P. KRISHNAN, "Optimal prefetching via data compression," in *Proceedings 32nd Ann. Symp. on Foundations of Computer Science,'* IEEE Computer Society, Los Angeles, Calif., 1991, 121-130.