Linear Waste of Best Fit Bin Packing on Skewed Distributions

Claire Kenyon,[1,*] Michael Mitzenmacher[2,†]
[1] *Université Paris-Sud, Paris, France; e-mail: kenyon@lri.fr*
[2] *Harvard University, Cambridge, MA; e-mail: michaelm@eecs.harvard.edu*

Received 17 April 2001; accepted 8 February 2002

DOI 10.1002/rsa.10037

ABSTRACT: We prove that Best Fit bin packing has linear waste on the discrete distribution $U\{j, k\}$ (where items are drawn uniformly from the set $\{1/k, 2/k, \ldots, j/k\}$) for sufficiently large k when $j = \alpha k$ and $0.66 \leq \alpha < 2/3$. Our results extend to continuous skewed distributions, where items are drawn uniformly on $[0, a]$, for $0.66 \leq a < 2/3$. This implies that the expected asymptotic performance ratio of Best Fit is strictly greater than 1 for these distributions. © 2002 Wiley Periodicals, Inc. Random Struct. Alg., 20: 441–464, 2002

1. INTRODUCTION

1.1. Background and Results

In the bin packing problem, one is given a sequence $L_n = a_1, \ldots, a_n \in (0, 1]$ of items and asked to pack them into bins of unit capacity so as to minimize the number of bins used. This problem is well known to be NP-hard, and a vast literature has developed around the design and analysis of efficient approximation algorithms for it. The most widely studied among these is the Best Fit algorithm, in which the items are packed on-line, with each successive item going into a partially filled bin with the smallest

Correspondence to: C. Kenyon
*Part of this work was done while the author was visiting AT&T.
†Supported in part by an Alfred P. Sloan Research Fellowship and NSF CAREER Grant CCR-9983832. Part of this work was done while the author was visiting AT&T.

residual capacity large enough to accommodate it; if no such bin exists, a new bin is started.

Best Fit was first analyzed in the worst case in 1974 in [8], where it was proved that the number of bins used is always within a factor 1.7 of the optimal number, so that the asymptotic performance ratio of Best Fit is 1.7. For the uniform distribution on [0, 1], the expected asymptotic performance ratio of Best Fit is 1, and more precisely the expected waste of Best Fit is $\Theta(n^{1/2}\log^{3/4}n)$ [15, 11]; here the waste is the total unused space, i.e., the difference between the number of bins used and the sum of the sizes of all the items.

To better understand Best Fit, researchers then turned their attention to the skewed distributions $U(0, a)$, where the item sizes are independent random variables uniform over the interval $[0, a]$ for some a strictly less than one.[1] For these distributions, the optimal packing is perfect in the sense that the limit of

$$E(OPT(L_n)/(a_1 + a_2 + \cdots + a_n))$$

equals 1 as n goes to infinity. Therefore, the expected asymptotic performance ratio of Best Fit is strictly greater than 1 if and only if the waste grows linearly in the number of items. Based on experimental evidence, it was conjectured that for all skewed distributions $U(0, a)$ the growth of the waste is linear [5, 1].

The discrete distributions $U\{j, k\}$ were introduced in [4] in the hope of gaining insight into the continuous case. Under distribution $U\{j, k\}$, items are drawn independently and uniformly from the set $\{1/k, 2/k, \ldots, j/k\}$. The distributions $U\{j, k\}$ approximate the continuous distribution $U(0, a)$ if one sets $j = ak$ and lets k go to infinity. Note that $U\{j, k\}$ can equivalently be thought of as the bins having capacity k and the item sizes being uniformly distributed on the integers $\{1, \ldots, j\}$; we generally use this formulation. Several extreme cases have been analyzed under $U\{j, k\}$: When $j = k - 1$, the expected waste is $\Theta(n^{1/2}\log k)$ [4]; when $j = k - 2$, the expected waste is $O(1)$ [10] (a result which can also be extended to First Fit [2]); and the expected waste is also $O(1)$ when $j \leq \sqrt{2k + 2.25} - 1.5$ [4]. The only specific cases where the expected waste of Best Fit has been proven to be linear is for the two distributions $U\{8, 11\}$ and $U\{9, 12\}$ [6]. Some other specific cases of $U\{j, k\}$ have been shown to have bounded expected waste [6]. Unfortunately, none of these results gave any information about the continuous distributions $U(0, a)$.

In this paper, we first study the discrete distributions $U\{j, k\}$ and prove that Best Fit has linear waste when k is large enough and $0.66k \leq j < (2/3)k$. We then proceed to prove our main result: Best Fit has linear waste for the continuous distributions $U(0, a)$ with $0.66 \leq a < 2/3$. This work therefore provides the first proof of linear waste for Best Fit under skewed continuous distributions.

1.2. Proof Techniques

In the discrete case, as in most previous work, we view the algorithm as a multidimensional Markov chain [6]. The states of the chain are non-negative integer vectors $s = (s_1, \ldots, s_{k-1})$, where s_i represents the current number of open bins of residual capacity i. It is a simple matter to write down the new vector s' that results from the arrival of any

[1]Next Fit was analyzed under $U(0,a)$ by Karmakar in 1982 [9, 7] using completely different techniques.

item $i \in \{1, \ldots, j\}$ when in state s. This defines an infinite Markov chain on $\mathbf{Z}_+^{k-1}$. The expected waste of Best Fit is directly related to the asymptotic behavior of this chain, which we analyze in detail.

The first novel ingredient in this paper is Lemma 2, a simple but crucial observation which we formulated after examining detailed simulations: If the maximum item size j is less than 2/3 times the bin capacity k, then Best Fit has at most one open bin with remaining capacity in the range $[k/3, k - j)$. (Strictly speaking the $k/3$ above is $\lceil k/3 \rceil$; we assume 3 divides k henceforth for simplicity.) Hence we focus on values of j with $j < (2/3)k$.

At a high level, our approach is simple. Our goal is to show that s_1, the number of bins of residual capacity 1, grows linearly in n. For most configurations, the next incoming item on average tends to increase s_1, or at least does not decrease it. The only exceptions are configurations for which the residual capacities of open bins are exactly $\{1, 2, \ldots, m\}$: There exist open bins of every remaining capacity $1, 2, \ldots, m$, and there exist no bins of larger remaining capacity (up to $j + 1$). Thanks to Lemma 2, this implies $m \leq k/3$. Intuitively, such configurations are then extremely short-lived, and inserting a few more items then typically modifies them into configurations in which s_1 is biased towards increasing. Thus the undesirable effects of these configurations should be amortized by running the Markov chain forward in time for a few steps.

In practice, running the Markov chain for C steps, there are j^C possible sequences to analyze, which would be computationally infeasible as j gets large. To get around this problem, our detailed analysis partitions the configurations into a constant number of groups. We then use stochastic domination, i.e., take the worst case configuration within each group. This worst case configuration is determined by dynamic programming: We successively find the worst configuration within each group given that there is one more item to be inserted, and from that calculate the worst case given that there are two more items to be inserted, and so on. This use of dynamic programming is commonly used in the analysis of Markov decision processes (see, e.g., [3, 13, 14]) and has been used for contention resolution protocols [12]; as far as we know, this is the first time it has been applied to stochastic bin packing. Although the derived dynamic program only has to deal with a constant number of cases, it is too large to be processed manually, and so we ended up writing a computer program for it. The actual table filled in by the program has tens of thousands of entries.

The continuous model follows the same general argument. The result for discrete distributions cannot be applied directly to continuous distributions by simply letting k go to infinity, for the following reason: When bin sizes are scaled to 1, our discrete result shows that s_1, the number of bins with remaining capacity $1/k$, grows linearly. However, as $1/k$ goes to 0 as k goes to infinity, the total contribution to the waste may still be sublinear. Hence in the continuous case, instead of studying s_1, we focus on bins with remaining capacity in the range $(0, \epsilon]$ for some small constant ϵ, and suitably adapt the proof of the discrete case.

2. ANALYSIS OF DISCRETE SKEWED DISTRIBUTIONS

2.1. The Markov Chain

In this section we study the discrete distribution $U\{j, k\}$ where the bin capacity is k and items are picked uniformly at random from the set $\{1, 2, \ldots, j\}$. We focus specifically on the case where $j = \alpha k$ for $0.66 \leq \alpha < 2/3$, and we assume that j and k are sufficiently large so that our arguments hold throughout.

Let us first recall the associated Markov chain setting. We shall denote the state of the system at time t by $s(t) = (s_1(t), \ldots, s_{k-1}(t))$, where $s_i(t)$ is the number of open bins at time t with residual capacity exactly i. Initially, the state of the system is $s(0) = (0, \ldots, 0)$, reflecting the fact that there are no open bins. Note that we often use s_i in place of $s_i(t)$ where the meaning is clear. Let ℓ be the size of the next item inserted. Let i be the smallest index such that $i \geq \ell$ and $s_i(t) > 0$, if such exists: In this case, the algorithm inserts item ℓ into a bin with capacity i, so we have $s_i(t + 1) = s_i(t) - 1$ and, if $i > \ell$, $s_{i-\ell}(t + 1) = s_{i-\ell}(t) + 1$; all other components of $s(t)$ are unchanged. If no such i exists, then the algorithm inserts item ℓ into an empty bin, so we have $s_{k-\ell}(t + 1) = s_{k-\ell}(t) + 1$ and all other components of s are unchanged. This completes the description of the Markov chain.

2.2. The Difficult Configurations

Our attack for proving instability is straightforward: We show that s_1, the number of almost full bins, is biased upward and hence tends to increase. Let $X(t)$ denote the number of ways to increase s_1 and $Y(t) \in \{0, 1\}$ denote the number of ways to decrease s_1. Define $W(t) = X(t) - Y(t)$. The value s_1 increases exactly when an item of size x is inserted and we have $s_x = 0$, $s_{x+1} \neq 0$, and so $X(t)$ is exactly the number of such pairs (s_x, s_{x+1}) with $x \leq j$. At every time step, if $s_1(t) = 0$ we have $Y(t) = 0$, and if $s_1(t) \neq 0$, then $Y(t) = 1$: Namely, s_1 can decrease only when an item of size 1 arrives. The only situations where s_1 is biased downward are if and only if s_1 has one way to decrease and no way to increase, i.e., if for some m, $s_1, s_2, \ldots, s_m \neq 0$ and $s_{m+1} = \cdots = s_{j+1} = 0$. Equivalently, s_1 is biased downward when $W(t) = -1$. We call these configurations where $W(t) = -1$ *difficult configurations,* as handling them is the challenge of the problem.

The lemma below enables us to conclude that m must be at most $k/3$ in any difficult configuration.

2.3. Combinatorial Lemmas

In this subsection, we demonstrate that one cannot have more than one bin with remaining capacity within a rather large range. We call the resulting lemma the *open range lemma.* First we recall the following fact, which is a classical property of Best Fit.

Fact 1. *Any two open bins with remaining capacities g and g' must have $g + g' < k$.*

The following fact, easily derived from Fact 1, is well known:

Fact 2. *There can be only one open bin with remaining capacity greater than or equal to $k/2$.*

We prove the following useful lemma.

Lemma 1. *Let $k/3 \leq y \leq z \leq k$ and $S = s_y + s_{y+1} + \cdots + s_z$. Assume that $S \neq 0$. Then S can only increase through the creation of new bins.*

Proof. The proof is by contradiction. Take $S \neq 0$, and let $i \in \{y, \ldots, z\}$ be such that $s_i \neq 0$. Assume that S can increase by inserting some item x into a partially filled bin g (with $g > z$). Then from Fact 1 we have

$$i + g \leq k. \tag{1}$$

Since Best Fit chooses to place x in bin g and not in bin i, and $i < g$, it must be that x does not fit in bin i, that is,

$$x > i. \tag{2}$$

The resulting bin, of size $g - x$, causes S to increase, so we must have

$$g - x \geq y \geq k/3. \tag{3}$$

Summing Eqs. (1), (2), and (3) together, we get

$$k = (g - x) + x + (k - g) > k/3 + i + i \geq k,$$

a contradiction. ∎

The following lemma can be seen as an extension of Fact 1 that greatly simplifies our subsequent analysis.

Lemma 2 [Open Range Lemma]. *If the maximum item size j is strictly less than $2k/3$, then $s_{k/3} + \cdots + s_{k-j-1} \leq 1$.*

Proof. Note that initially $s_{k/3} + \cdots + s_{k-j-1} = 0$. Hence we need only show that when $s_{k/3} + \cdots + s_{k-j-1} = 1$, it cannot increase.

Consider any time t when $s_{k/3} + \cdots + s_{k-j-1} = 1$ and let $i \in \{k/3, \ldots, k - j - 1\}$ be such that $s_i = 1$. Let i' be such that $k/3 \leq i' \leq k - j - 1$. How can $s_{i'}$ increase (without decreasing s_i)? Note that $s_{i'}$ cannot increase by having an item of size $k - i'$ placed into an empty bin, since $k - i'$ is greater than or equal to $k - (k - j - 1) = j + 1$, one more than the largest item size. On the other hand, by Lemma 1, it cannot increase by adding some item to a partially filled bin. Hence Lemma 2 holds. ∎

We reiterate that the open range lemma simplifies the analysis, since it ensures that there is some well-defined range of values i where most of the values s_i must be 0, and hence that any difficult configuration with $s_1, s_2, \ldots, s_m \neq 0$ and $s_{m+1} = \cdots = s_{j+1} = 0$ must have $m \leq k/3$.

2.4. Reducing the Process to a Small Number of Steps

We first prove that as a consequence of Lemma 2, there exists a significant range adjacent to the open range and in which, some constant fraction of the time, there are no open bins.

Lemma 3. *There exists three constants* $\epsilon > 0$, $p > 0$, *and* T_0 *such that for every* $t \geq T_0$, *the probability* $\Pr\{s_{k-j}(t) = s_{k-j+1}(t) = \cdots = s_{k-j-1+\epsilon j}(t) = 0\}$ *is at least* p.

Proof. Let $\gamma = k - j - k/3$ denote the size of the open range. We study the transition from t to $t + 1$. For notational convenience, let $S = S(t)$ and $s_x = s_x(t)$.

Let $S = s_{k-j} + s_{k-j+1} + \cdots + s_{k-j-1+\epsilon j}$. Say that a configuration is "good" if $s_{k-j-1} = s_{k-j-2} = \cdots = s_{k-j-\gamma/4} = 0$, and "bad" otherwise.

With each configuration, we associate a potential function Φ equal to i if $S = i$ and the configuration is good, and equal to $i + 1/2$ if $S = i$ and the configuration is bad. We show that Φ is biased downward.

Assume that $\Phi \geq 1$, so that $S \neq 0$. There are two cases.

In the first case, consider a good configuration. From Lemma 1, S can only increase by direct creation of new bins. Hence there are at most ϵj ways to increase S, so S increases with probability at most ϵ. The new configuration is still good. S decreases whenever an item of size in $\{k - j - 1, k - j - 2, \cdots, k - j - \gamma/4\}$ is inserted (and the new configuration is still good). Hence S decreases with probability at least $\gamma/(4j)$. The configuration changes from good to bad without changing S with probability at most $\gamma/(4j)$. Thus the new value Φ' of the configuration after insertion satisfies

$$E(\Phi' - \Phi) \leq \epsilon \times (+1) + \gamma/4j \times (-1) + \gamma/4j \times (+1/2) = \epsilon - \gamma/8j,$$

which is negative for ϵ small enough.

In the second case, consider a bad configuration. From Lemma 1, S can only increase by direct creation of new bins. Hence there are at most ϵj ways to increase S, so S increases with probability at most ϵ. The new configuration is still good. The configuration changes from bad to good without increasing S with probability at least $3\gamma/4$, the size of the part of the open range which is known to be empty. Thus the new value Φ' of the configuration after insertion satisfies

$$E(\Phi' - \Phi) \leq \epsilon \times (+1) + 3\gamma/4j \times (-1/2) = \epsilon - 3\gamma/8j,$$

which is negative for ϵ small enough.

Thus, for every configuration such that $\Phi \geq 1$, we have shown that Φ is biased downward. This implies boundedness of Φ: There are constants C and T_1 such that, for every $t \geq T_1$, $E(\Phi(t)) \leq C$. Applying Markov's inequality, for every $t \geq T_1$, with probability at least $1/2$, $\Phi(t) \leq 2C$, i.e., with positive probability the state is at most some constant number of steps from a state where $S = 0$. From this state, a state where $S = 0$ can be reached in a constant number of steps with constant probability. Thus there exists a constant T_0 such that, for every $t \geq T_0$, the probability that $S(t) = 0$ is greater than some positive constant. ∎

Recall that we define $X(t)$ to be the number of ways for s_1 to increase, or equivalently, the number of pairs $s_x = 0$, $s_{x+1} \neq 0$, for $1 \leq x \leq j$. We define $Y(t)$ to be the number of ways for s_1 to decrease, or equivalently, the indicator function of the event $s_1(t) \neq 0$. We define $W(t) = X(t) - Y(t)$. The instances where $W(t) = -1$, or difficult configurations, play a key role in our subsequent proof. Indeed, our goal is to show that we can amortize the downward bias in s_1 caused by the difficult configurations, by considering the bin packing Markov chain for several steps when we hit a difficult configuration. The next few lemmas provide the technical framework that allow us to conclude that this amortization approach applies. Subsequently, we provide a dynamic programming formulation to demonstrate that the bias starting from a difficult configuration can indeed be made up for over the next several steps.

Lemma 4. *Let p, ϵ and T_0 be as in Lemma 3. Then for every $t \geq T_0 + 3$, we have*

$$\Pr\{W(t) \geq 1\} \geq p\epsilon^3/2.$$

Proof. Assume that at time t, $s_{k-j} = s_{k-j+1} = \cdots = s_{k-j-1+\epsilon j} = 0$. From Lemma 3, this has probability at least p. We examine the probability that $W(t+3) \geq 1$, using three cases.

- If the configuration at t has $s_\ell = 1$ for some $\ell \in \{j - \epsilon j + 1, \ldots, j\}$, then from Fact 2 that value is unique. Inserting first an item in the range $\{j - 2\epsilon j + 1, \ldots, j - \epsilon j\}$ and then two distinct nonadjacent items in the range $\{j - \epsilon j + 1, \ldots, j\}$ creates two nonadjacent coordinates equal to 1 among $s_{k-j}, \ldots, s_{k-j-1+\epsilon j}$. This raises $X(t+3)$ to at least 2 and $W(t+3)$ to at least 1. This happens with probability at least $(\epsilon j/j) \times (\epsilon j/j) \times ((\epsilon j - 3)/j) \geq \epsilon^3/2$.
- If the configuration at t has $s_\ell = 1$ for some $\ell > j$, then inserting first an item in the range $\{j - \epsilon j + 1, \ldots, j\}$ removes that bin of remaining capacity ℓ. Then inserting two items in the range $\{j - \epsilon j + 1, \ldots, j\}$ increases two coordinates by 1 among $s_{k-j}, \ldots, s_{k-j-1+\epsilon j}$. If these two coordinates are distinct and nonadjacent, this raises $X(t+3)$ to at least 2 and $W(t+3)$ to at least 1. This happens with probability at least $(\epsilon j/j) \times ((\epsilon j)/j) \times ((\epsilon j - 3)/j) \geq \epsilon^3/2$.
- If the configuration at t has $s_\ell = 0$ for every $\ell > j - \epsilon j$, then inserting three distinct non adjacent items in the range $\{j - \epsilon j + 1, \ldots, j\}$ creates three nonadjacent coordinates equal to 1 among $s_{k-j}, \ldots, s_{k-j+\epsilon j}$. This raises $X(t+3)$ to at least 3 and $W(t+3)$ to at least 2. This also has probability at least $\epsilon^3/2$.

Thus in all cases at time $t + 3$ we have $W(t) \geq 1$ with probability at least $\epsilon^3/2$, which proves the lemma. ■

Lemma 5. *Let Σ denote the set of initial configurations such that $W(0) = -1$, and assume that for some constant C,*

$$\inf_{s \in \Sigma} E(W(0) + W(1) + \cdots + W(C-1) \big| s \text{ at time } 0\} = b > 0. \tag{4}$$

Then the expected number of bins with remaining capacity 1 grows linearly in t.

Proof. Let $Q(T)$ be a random variable representing the number of times over the first T steps that the configuration of the random process is in Σ.

The distribution of $s_1(t)$ given the state at time $t - 1$ is

$$s_1(t) = \begin{cases} s_1(t-1) + 1 & \text{w.p. } X(t-1)/j, \\ s_1(t-1) - 1 & \text{w.p. } Y(t-1)/j. \end{cases}$$

Thus $E(s_1(t)|\text{state at } t-1) = s_1(t-1) + W(t-1)/j$, and summing gives $E(s_1(T)) = E(\sum_0^{T-1} W(t))/j$.

We always have $W(t) \geq -1$. The difficult configurations are exactly the ones for which $W(t) = -1$, i.e. the configuration of Σ. Consider running the chain for n steps, and divide time into *supersteps.* A superstep is simply a normal step of the chain, except in the case where we reach a difficult configuration where $W(t) = -1$. In this case, we consider the evolution of the system over C steps for some large enough constant C, and all the steps from this point t until the time $t + C - 1$ are combined into a superstep; in fact, we call this a *long superstep.* Every short superstep has $W(t) \geq 0$. With this framework, we compute two lower bounds on $E(s_1(T))$

- There are at least $Q(T)/C$ long supersteps in $[0, T]$, and so

$$
\begin{aligned}
E(s_1(T)) &= E\left(\sum_{t<T} \frac{W(t)}{j}\right) \\
&\geq E\left(\sum_{t\leq T, t\in \text{long superstep}} \frac{W(t)}{j}\right) \\
&\geq \frac{1}{j}\frac{E(Q(T))}{C}b \qquad \text{by the lemma's assumption.}
\end{aligned}
$$

- But also,

$$
\begin{aligned}
E(s_1(T)) &= E\left(\sum_{t<T} \frac{W(t)}{j}\right) \\
&\geq \frac{1}{j}\sum_{t<T} \Pr\{W(t) \geq 1\} - \frac{1}{j}E(Q(T)) \\
&\geq \frac{1}{j}\left[\frac{(T-T_0)p\epsilon^3}{2} - E(Q(T))\right] \\
&= \frac{1}{j}\left(\frac{pT\epsilon^3}{2 - E(Q(T))}\right) - \frac{T_0}{j}p\frac{\epsilon^3}{2}.
\end{aligned}
$$

The expression corresponding to the maximum of these two lower bounds is $\Omega(T)$. This can most easily be seen by noting that multiplying C/b by the first lower bound and adding the second lower bound yields an expression that is $\Omega(T)$. Hence, the expected number of bins with remaining capacity 1 grows linearly in t. ∎

The next lemma handles some of the technical considerations in working over multiple steps.

Lemma 6. *With any configuration $s \in \Sigma$, one can associate an integer m such that s_1, $s_2, \ldots, s_m > 0$ and $s_{m+1} = s_{m+2} = \cdots = s_{j+1} = 0$. If there exist positive constants C, K, and ϵ such that for all $k > K$ and for every $s \in \Sigma$,*

$$
\frac{1}{C}E\left(\sum_{0\leq t<C} S(t)|s \text{ at time } 0, \text{ no items inserted from } U\right) > 1 + \epsilon, \tag{5}
$$

where $S(t) = s_{m+1}(t) + \cdots + s_{j+1}(t)$ *and* $U = \{m, m-1, \ldots, m-C+1\}$, *then Eq.* (4) *holds.*

Proof. Assume that at time 0 we are in a difficult configuration s, and let $S(t) = s_{m+1}(t) + \cdots + s_{j+1}(t)$. Obviously, $S(0) = 0$, $S(t) \le t$, and $\sum_{t<C} S(t) \le C^2/2$. Let $U = \{m, m-1, \ldots, m-C+1\}$. We have:

$$\begin{aligned}
E\left(\sum_{t<C} S(t)\right) &\ge \Pr\{\text{no insertion from } U\} \times E\left(\sum S(t) \middle| \text{no insertion from } U\right) \\
&= \left(1 - \frac{C}{j}\right)^C E\left(\sum S(t) \middle| \text{no insertion from } U\right) \\
&\ge E\left(\sum S(t) \middle| \text{no insertion from } U\right) - O\left(\frac{C^2}{j}\right) \times \frac{C^2}{2} \\
&= E\left(\sum S(t) \middle| \text{no insertion from } U\right) - O\left(\frac{C^4}{j}\right).
\end{aligned}$$

Now, consider the following events.

- A_1. The coordinate s_{m+1} is positive.
- A_2. For some $a > m$, the coordinate s_a is greater than or equal to 2.
- A_3. For some $a > m + 1$, the coordinates s_a and s_{a-1} are both positive.

Remarking that, at any time $t \in \{0, \ldots, C-1\}$, there are at most $C - 1$ values $s_x \neq 0$ with $x > m$, we see that the probability that event A_i (for $i = 1$, 2 or 3) first occurs at time $t + 1$ is at most C^2/j; thus the probability that at least one of the events A_1, A_2 and A_3 occurs at some time $t \le C - 1$ is bounded by $C \times 3C^2/j = O(C^3/j)$.

Observe that if none of A_1, A_2, A_3 hold, then $X(t) \ge S(t)$. Since $S(t) < C$ and $X(t) < C$ for all $t < C$, we deduce

$$\begin{aligned}
E\left(\sum_{0\le t<C} X(t)\right) &\ge E\left(\sum_{0\le t<C} X(t) \middle| \bar{A}_1, \bar{A}_2, \bar{A}_3\right) \times \Pr(\bar{A}_1, \bar{A}_2, \bar{A}_3) \\
&\ge E\left(\sum_{0\le t<C} X(t) \middle| \bar{A}_1, \bar{A}_2, \bar{A}_3\right) - O(C^4/j) \\
&\ge E\left(\sum_{0\le t<C} S(t) \middle| \bar{A}_1, \bar{A}_2, \bar{A}_3\right) - O(C^4/j) \\
&= E\left(\sum_{0\le t<C} S(t)\right) - O(C^4/j).
\end{aligned}$$

Thus

$$
\begin{aligned}
E\left(\sum W(t)\right) &\geq E\left(\sum X(t)\right) - C \\
&\geq E\left(\sum S(t)\right) - C - O(C^4/j) \\
&\geq E\left(\sum S(t) \middle| \text{no insertion from } U\right) - C - O(C^5/j) \\
&\geq (1 + \epsilon)C - C - O(C^5/j) \\
&= \epsilon C - O(C^5/j) \\
&\geq \epsilon C/2
\end{aligned}
$$

for j large enough, hence the lemma. ∎

Thus we have reduced the problem to proving Eq. (5). The remainder of the section is devoted to proving this equation.

Our analysis of $S(t)$ uses stochastic domination. The domination argument is only valid as long as the following event does not occur.

- A_4. The coordinate s_m is equal to 0.

Since during $\{0 \ldots C - 1\}$ there is no insertion of any item among $U = \{m, m - 1, \ldots, m - C + 1\}$, we are guaranteed that event A_4 does not occur.

Note that we require that the right-hand side of Eq. (5) to be greater than $1 + \epsilon$ so as to absorb the error terms introduced by events A_1, A_2, A_3, A_4 for sufficiently large k.

Given that the dynamic program below allows us to conclude that Eq. (5) holds, we have the following theorem.

Theorem 1. *The number of s_1 bins for Best Fit bin packing under the discrete distribution $U\{j, k\}$, $33k/50 < j < 2k/3$, grows linearly in n for sufficiently large k.*

Proof. Given that Eq. (5) holds, Lemmas 5 and 6 allow us to conclude the theorem. ∎

2.5. Describing Configurations with a Small Number of Parameters

We assume that $j = \alpha k$ for $\alpha \in (0.66, 2/3)$. We begin in a state where $s_1, s_2, \ldots, s_m \neq 0$ and $s_{m+1} = \cdots = s_{j+1} = 0$. Our proof is based on stochastic domination.

So far, we have reduced ourselves to studying the Markov chain over a small number of steps. The space of configurations is still high dimensional, since it sits in $\mathbf{Z}^{k+1}$. We now introduce four parameters that are sufficient for the analysis.

The first parameter is m, such that in the initial configuration $s_1, \ldots, s_m \neq 0$ and $s_{m+1} = \cdots = s_{j+1} = 0$.

The presence of bins which are less than half full is important for the behavior of the process. To see why they have a significant impact, consider the case where $j = 0.64k - 2$, and $m = 0.32k$, and $s_{0.64k} = 1$. In this case it is impossible for any of s_{m+1}, $s_{m+2}, \ldots, s_{k/2}$ to increase on the next step. Thus it is important to keep track of the presence of bins of remaining capacity greater than $k/2$. Remember from Fact 2 that there can be only one such bin.

Let us call a nonempty bin with remaining capacity greater than $k/2$ a *light* bin. Fact 2 says there is at most one light bin. Hence we can consider two kinds of states: those with a light bin, and those without. In fact, we refine our analysis further, to three kinds of states, by dividing states with a light bin into two subtypes. Let us call a light bin *helpful* if its remaining capacity is at most $j + 1$. A helpful light bin with remaining capacity r can immediately lead to an increase in s_1, if an appropriately sized item arrives, since if $s_r > 1$ then s_{r-1} cannot be. Similarly, we call a light bin *unhelpful* if its remaining capacity is greater than $j + 1$.

Given m, we may now group the states of our Markov chain into *superstates,* where a superstate is a triple $(S(t), A(t), B(t))$ such that $S(t) = s_{m+1}(t) + \cdots + s_{j+1}(t)$, $A(t)$ is a 0/1 random variable representing whether or not there is a light helpful bin, and $B(t)$ is a 0/1 random variable representing whether or not there is a light unhelpful bin. Note that when $A(t) = 1$, we must have that $S(t) \geq 1$.

We also group the values of m into four classes, depending on whether m/k is in (0, 0.12], (0.12, 0.22], (0.22, 0.25], or (0.25, 1/3] (testing to find those ranges took some experimentation).

2.6. Transition Probabilities: Accuracy Issues

In the computer program that we use for studying the Markov chain, it is convenient to consider that the probability that an arriving item has value in the range $[a, b]$ is *exactly* $(b - a)/k$ for $b < j$ and is $(j - a)/k$ for $b \geq j$. This is acceptable because we are interested in the asymptotic behavior as k grows large in our analysis, so we are considering what is essentially the limiting case as k grows to infinity. This avoids artifacts in the analysis that depend on k, and for sufficiently large k the difference is a lower order term that can be absorbed in the ϵ, as stated in the following lemma.

Lemma 7. *Let $\tilde{M}$ be a Markov chain with the same configuration space as our Markov chain M, and such that every transition probability satisfies $\tilde{p}(s, s') = p(s, s')(1 + O(1/k))$. Let $U = \{m, m - 1, \ldots, m - C + 1\}$. Then, for any initial configuration s, and any constant C, we have*

$$E_{\tilde{M}}\left(\sum_{0\leq t<C} S(t) \middle| s \text{ at time 0, no items from } U\right)$$

$$= E_M\left(\sum_{0\leq t<C} S(t) \middle| s \text{ at time 0, no items from } U\right) + O(C^3/k).$$

Proof. The proof is straightforward and similar to arguments of the previous section: The two Markov chains only differ when the inserted item is at the boundary of $[a, b]$, and this event has probability $O(1/k)$ at every step, accumulating to $O(C/k)$ over C steps. When there is a mismatch the difference in the summations is $O(C^2/k)$, as in Lemma 6, since the difference in $S(t)$ is at most C for each of C steps. Hence the difference in the expectations is $O(C^8/k)$ which is negligible for C constant and k sufficiently large. ∎

2.7. The Dynamic Program

For convenience, in what follows let $C = 100$. (Any sufficiently large constant would do, and it turns out $C = 100$ suffices.) We now use a computer program to assist in computing successive lower bounds for the values of $E_c = E(\sum_{0 \le t < c} S(t))/c$ given the initial configuration and assuming that there are no inserted items from U, in order to prove that $E_C > 1 + \epsilon$ for some constant ϵ and for every difficult configuration $(S(0) = 0, A(0), B(0))$. Our methodology is to use dynamic programming.

We adopt the following notation: Let $F_c(S(0), A(0), B(0))$ be a lower bound on the value of E_c when the configuration at time 0 is in the superstate $(S(0), A(0), B(0))$, where, for the purpose of inductive reasoning, $S(0)$ can be any integer between 0 and C. The start of the induction is $F_1(S(0), A(0), B(0)) = S(0)$. We then successively compute lower bounds $F_2, F_3, \ldots, F_C$ using dynamic programming, where for F_x we obtain the value for all starting states $(S(0), A(0), B(0))$ with $S(0) \le C - x + 1$. To compute F_{x+1} from F_x, we consider all possible first steps of the chain.

For intuition, let us temporarily assume (for this paragraph) that from an initial state $(S(0), A(0), B(0))$ there is a set of superstates Q that can be reached in one step, and the probability of moving to a superstate $q \in Q$ is exactly p_q, regardless of which configuration of the superstate $(S(0), A(0), B(0))$ we start from. Then we have the simple calculation

$$(c + 1)F_{c+1}(S(0), A(0), B(0)) = c \sum_{q \in Q} F_c(q) p_q. \tag{6}$$

Unfortunately, since superstates are sets of configurations, not one single configuration, p_q is not well defined in general. For example, suppose $A(t) = 1$, and an item with size in the range $[0.5k, j]$ arrives. Such an item might be placed in the light bin; alternatively, such an item might prove too large for the light bin, and instead cause $S(t)$ to increase. The effect of the item depends on the exact residual capacity of the light bin and on the exact value of the entering item (in other cases the exact value of m is also relevant).

To deal with such ambiguous situations, we use a worst-case analysis. This is similar in spirit to allowing an oblivious adversary some limited power in deciding the flow of the process. In such ambiguous situations we can compute lower bounds for F_{c+1} from our lower bounds on F_c by taking the *worst-case possibility* whenever a choice occurs. That is, we determine the configuration that produces the set of values p_q for Eq. (6) that minimize the values for F_{c+1}. This allows us to keep our simpler state space, which is important for keeping our analysis manageable. However, it complicates the analysis, in that we must consider many possibilities at each step. For this reason we use computer analysis.

Note that we require a worst case analysis both because of the ambiguity caused by the grouping into superstates, and because of the grouping of m into four types. To summarize, with $S(0)$ bounded by 100, four types for m, and two binary 0/1 variables, and only 100 steps to consider, we have reduced the problem to constant size. Hence our calculations, when aided by a machine, are actually relatively straightforward.

Let us detail, for example, one of the cases for which our program does a worst-case analysis. (It is the first subcase of the first case in the Appendix.) Assume that m/k is in the range $[.25, 1/3]$ and that the initial configuration has no light bin, i.e., $A = B = 0$ initially. Let wk denote the value of the first inserted item.

1. If $w < 1/4$, then $wk < m$ and so the item is placed in a bin of remaining capacity wk, so that S does not change and no light bin is created.
2. If $1/4 \leq w \leq 1/3$, then
   - either the item fits into some existing bin in the range $\{1/4, \ldots, m\}$, so that S does not change and no light bin is created;
   - or the item fits into some existing bin in the range $\{m + 1, \ldots, \lceil k/2 \rceil - 1\}$; this reduces the bin's capacity to less than m, so S is decreased by 1;
   - or the item does not fit into any existing bin in the range $\{m + 1, \ldots, \lceil k/2 \rceil - 1\}$; then a new light bin is opened, and its residual capacity is $k(1 - w) > j + 1$, so it is an unhelpful bin and B is set to 1.
3. If $1/3 \leq w \leq 1 - \alpha$, then
   - either the item fits into some existing bin in the range $\{m + 1, \ldots, \lceil k/2 \rceil - 1\}$, so that the bin's capacity is reduced to less than m; then S is decreased by 1;
   - or the item does not fit into any existing bin in the range $\{m + 1, \ldots, \lceil k/2 \rceil - 1\}$; then a new light bin is opened, and its residual capacity is $k(1 - w) > j + 1$, so it is an unhelpful bin and B is set to 1.
4. If $1 - \alpha \leq w \leq 1/2$, then
   - either the item fits into some existing bin in the range $\{m + 1, \ldots, \lceil k/2 \rceil - 1\}$, and this reduces the bin's capacity to less than m; then S is decreased by 1;
   - or the item does not fit into any existing bin in the range $\{m + 1, \ldots, \lceil k/2 \rceil - 1\}$; then a new light bin is opened, and its residual capacity is $k(1 - w)$, where $k/2 \leq k(1 - w) \leq j + 1$; therefore, it is a helpful light bin, so that S increases by 1 and A is set to 1.
5. If $1/2 \leq w \leq \alpha$, then a new bin is created, of residual capacity less than 1/2, and so S is increased by 1.

The other cases are similar and detailed in the Appendix. When given an option between two transitions, the dynamic program chooses the worst transition.

We remark that we have chosen the barrier 0.66 for convenience, and we have not tried to determine the exact range for which the argument holds. It appears that additional work detailing the cases would be required, however, to extend the lower bound of the range below $0.65 = 13/20$.

Finally, our case analysis is based on choosing a specific value of α. To claim that the proof works for the entire range of α from [0.66, 2/3), there are two possibilities. First, we could loosen our case structure to allow both m and α to vary. This is possible but complicates the already extensive case analysis. We suggest another approach that we have taken. Suppose one focuses on a specific value of α (such as $\alpha = 0.66$), and proves $E_C > 1 + \epsilon$ by splitting up the possible values of m over a small number of ranges. We claim that then there are small constants $\delta, \epsilon' > 0$ such that $E_C > 1 + \epsilon'$ for $\alpha' \in [\alpha - \delta, \alpha + \delta]$. Assuming this is true, it suffices to try a sufficiently dense subset of α values in the range [0.66, 2/3), and by the "continuity" implied by the above argument, we may conclude $E_C > 1 + \epsilon$ for a suitable ϵ everywhere in the interval. Note again that the upper limitation of 2/3 is due to the structural limitations of the open range lemma.

Lemma 8. *Suppose that for a specific value of* α *and some* $\epsilon > 0$ *we have* $E_C > 1 + \epsilon$*. Then there are constants* δ, $\epsilon' > 0$ *such that for all* $\alpha' \in [\alpha - \delta, \alpha + \delta]$, $E_C > 1 + \epsilon'$.

Proof. The natural approach is to consider any starting state, and note that the two chains diverge over the C steps only if an item of size between α and α' enters. Hence the probability the chains deviate over C steps is $O(C\delta)$, from which we may conclude similarly to Lemma 7 that if $W(0) = -1$ the value of E_C for the two chains can differ only by $O(C^3\delta)$. This approach suffers the drawback that the valid states when the bin packing process is run with $j = \alpha k$ are not the same as the valid states when the bin packing process is run with $j = \alpha' k$; indeed, the open range lemma shows the valid states in each setting are different.

Our simplified process with state $(S(t), A(t), B(t))$ derived from the Markov chain can be used in a similar fashion while avoiding this problem. Examining the dynamic programming framework given in the Appendix shows that the lower bound we derived for F_c behaves the same over C steps regardless of whether $j = \alpha k$ or $j = \alpha' k$, unless some incoming item has weight between α and α' or between $1 - \alpha$ and $1 - \alpha'$. Hence the probability that two lower bounds deviate over C steps is $O(C\delta)$. The value of E_C can therefore differ only by an amount that is $O(C^3\delta)$ between the two situations $j = \alpha k$ or $j = \alpha' k$ (similar to Lemma 7). By choosing the value of δ to be a sufficiently small constant, this difference can be made less than ϵ, proving the lemma. ∎

3. ANALYSIS OF CONTINUOUS SKEWED DISTRIBUTIONS

Our above analysis can easily be extended to show that linear waste occurs when bins have size 1 and the items are uniform over the real interval $(0, a]$, for a in the range $0.66 \le a < 2/3$. We believe this is the first nontrivial probabilistic analysis of Best Fit for the continuous case when the interval is other than (0, 1].

The result of Theorem 1 cannot be extended immediately as k grows to infinity to yield the continuous case, since if we scale back the bin sizes in the $U\{j, k\}$ model so that all bins have size 1, Theorem 1 only says that the number of bins with remaining capacity $1/k$ grows linearly with n. If both n and k are growing to infinity, it is not clear that this implies divergence of the waste. A more careful argument avoids this, giving us that for sufficiently large k, the expected waste for $U\{j, k\}$, where $j = \alpha k$ grows at least linearly in n with a constant factor that is independent of k. Rather than extend the discrete case, however, we simply outline the corresponding argument for the continuous case.

We first note that the open range lemma that simplifies our analysis has a continuous analog.

Lemma 9. *Let* $1/2 < a < 2/3$. *Then when the bin capacity is 1 and the item sizes are drawn from* $(0, a]$, *there can only be one bin with remaining capacity in the interval* $(1/3, 1 - a)$.

Moving from the discrete case to the continuous case requires some care. The primary difference is that for the continuous case, we consider the creation and deletion of bins with remaining capacity $[\delta/\gamma, \delta]$ for some suitably small constant δ and suitably large constant γ. We call the number of such bins $s_{\delta,\gamma}$. Note that the choices of δ and γ are dependent on a. Also note that we consider bins with remaining capacity only in the range

$[\delta/\gamma, \delta)$ rather than $[0, \delta)$, because if we allow the lower bound to go to 0, we cannot give adequate lower bounds for the waste represented by these bins. With the remaining capacity of the bin bounded below by a suitably small constant, we avoid this problem.

We sketch the proof, which follows the same outline as before, in that we use our analysis of the superstate $(S(t), A(t), B(t))$. Whereas before $S(t)$ represented the number of nonempty bins that could possibly increase s_1 on the next step, it now represents the number of nonempty bins that could possibly increase $s_{\delta,\gamma}$ over the next step. Moreover, we restrict further that in $S(t)$ we only count bins that increase $s_{\delta,\gamma}$ *uniformly,* in the following sense: The distribution of the remaining capacity when an item is placed in a bin that contributes to $S(t)$ is uniform over the range $[\delta/\gamma, \delta)$. As a specific example, suppose that there is a bin with remaining capacity $x < a + \delta$, and no bin with remaining capacity in the range $(x - \delta, x - \delta/\gamma]$. Then this bin represents a way that $s_{\delta,\gamma}$ can increase, as any incoming item in the range $(x - \delta, x - \delta/\gamma]$ increases $s_{\delta,\gamma}$. Moreover, since the distribution of an incoming item conditioned on its being from the interval $(x - \delta, x - \delta/\gamma]$ is uniform, it is clear that the resulting remaining capacity of the bin that results is uniform over the range $[\delta/\gamma, \delta)$. Assuming without loss of generality that all open bins have distinct remaining capacities, $S(t)$ is simply the number of distinct values x with this property at time t. The values $A(t)$ and $B(t)$ then have similar meanings as in the discrete case.

The difficult cases occur when $S(0) = 0$. We can show with the same argument as in the discrete case that even when $S(0) = 0$, over the next C steps we have $E_C = E[\sum_{t=0}^{C-1} \frac{S(t)}{C} \mid \text{state at time } 0 > 1 + \epsilon'$ for some constants ϵ' and C. Indeed, we may use the same dynamic programming formulation as in the discrete case, where the evolution of $(S(t), A(t), B(t))$ is analyzed independently of k. As described in Section 2.6, our Markov chain analysis already describes the limiting case as k goes to infinity. By making δ sufficiently small and γ sufficiently large, the effect on the analysis of bins with extremely small remaining capacity can be made arbitrarily small, following the same continuity style argument as in Section 2.7.

As in the discrete case, there are complications to consider. For example, in the discrete case we did not wish to allow two bins with remaining capacities i and $i + 1$ to both count for $S(t)$; we noted that the probability of such an event was small, vanishing as k grew large. We do the equivalent here; with small probability, we may introduce over multiple steps of the process two bins with remaining capacities x and $x + \rho$ for $\rho < \delta$. In this case we do not wish to count both in $S(t)$. By making δ a sufficiently small constant we can make the probability of such an event arbitrarily small, and as in Lemma 6 the constant ϵ' covers the small error term introduced.

We may conclude from our dynamic programming computation that the rate at which bins with remaining capacity in the range $[\delta/\gamma, \delta)$ are created is slightly larger (by a small constant factor) than the rate at which they disappear. Hence the rate at which $s_{\delta,\gamma}$ increases is slightly larger than the rate at which it decreases. Since the remaining capacity of a bin contributing to $s_{\delta,\gamma}$ is at least the constant δ/γ, this implies linear waste.

Theorem 2. *Best Fit bin packing with item sizes chosen uniformly from the range* $[0, a]$ *has linear waste, for* $0.66 \leq a < 2/3$.

4. CONCLUSIONS AND OPEN PROBLEMS

We have shown linear waste for discrete distributions of the form $U\{j, k\}$, where $j = \alpha k$ for α in the range [0.66, 2/3), and from this derived an argument for the continuous case $(0, a]$ for a in the above range. Our analysis depends on a careful breakdown of the underlying Markov chain. It may be possible to simplify our argument substantially.

One open problem is to determine the extent of the range of values for which our argument functions. It is clear that the range [0.66, 2/3) can be expanded, in both directions. Our current argument appears to work down to values of α (or a, in the continuous case) of almost 0.65 = 13/20 and could perhaps be improved further. We believe improvements using the same techniques could be had by expanding the case-by-case analysis. The most obvious and tractable starting points are to increase the number of distinct ranges analyzed for the values of m/k and to enhance the state space by giving more detailed information about the location of any light bin. Extending the range upwards would be more delicate since Lemma 2 would no longer apply.

It seems likely that our techniques might also apply to First Fit. It is conjectured that First Fit has linear waste on input distributions $[0, a]$ for $a < 1$ as well. The First Fit argument will likely be more difficult, since for First Fit the order in which nonempty bins are created plays a role, but [2] demonstrates that it is possible to extend these types of Markov chain arguments to First Fit.

The question of proving linear waste for $U\{j, k\}$ when j differs from k by a suitably large constant remains tantalizingly unsolved.

APPENDIX: CASE ANALYSIS

The following presents our case analysis for the dynamic programming problem. We recall the framework. Here $j = \alpha k$, where $0.66 \leq \alpha < 2/3$. We consider separate ranges of the value m (where $s_m \neq 0$ and $s_{m+1} = 0$). The entering item has weight wk, where $0 \leq w \leq \alpha$. The value $S(t) = s_{m+1} + \cdots + s_{j+1}$ corresponds to the number of ways to increase s_1; $A(t)$ is 1 if and only if there is a helpful light bin, and 0 otherwise; $B(t)$ is 1 if and only if there is an unhelpful light bin, and 0 otherwise.

Given an entering item size, the state $(S(t), A(t), B(t))$ changes to a state $(S(t + 1), A(t + 1), B(t + 1))$, which may or may not be the same as the previous state. There may be several possible states $(S(t + 1), A(t + 1), B(t + 1))$ that can arise from $(S(t), A(t), B(t))$ on the introduction of an item of weight wk. In this case we determine the possible next states, with the model that the adversary may choose among them. In the notation of Section 2.7, given the values for F_c, in ambiguous situations we choose the move that minimizes the computed value for F_{c+1}.

In some cases we require restricting the adversary, in the following sense. If $S(t) = 1$, then there is one bin with remaining capacity at least m but at most $k/2$. The value of the remaining capacity may determine the possible transitions at the next time step, which therefore determines the calculation of the value F_{c+1} for that state. We may force the adversary to choose the value of the remaining capacity, and then consider the effect of all possible entering item sizes. The adversary can choose the worst possible remaining capacity at each step in the dynamic programming calculation, as this information is not recorded in the state, but it cannot change this value according to size of the next incoming item.

Since the dynamic programming computation for F_c is simply tracking the sum of the variable $S(t)$ over all time steps, in describing the cases we need only worry about changes in the state. Also, it becomes slightly less cumbersome to use $S' = S - A$ instead of S in what follows; that is, S' excludes any light bins (if any) from the count. We therefore use S', A, and B in place of $S(t)$, $A(t)$, and $B(t)$ to track the state, where the meaning is clear. We use the expression increase S' to mean that the value of S' increases by 1 during that transition. We use the expression decrease S' to mean that the value of S' decreases by 1 during that transition. Notice that if $S' = 0$ decreasing S' is not possible; in such a case the adversary must choose among the other possible choices for that transition.

- Case 1: $\frac{1}{4} \leq \frac{m}{k} \leq \frac{1}{3}$
  - Subcase: $A, B = 0$
    - If $w \leq 1/4$, make no change in the state.
    - If $1/4 \leq w \leq 1/3$, then decrease S', or make no change, or set B to 1.
    - If $1/3 \leq w \leq 1 - \alpha$, then decrease S' or set B to 1.
    - If $1 - \alpha \leq w \leq 1/2$, then decrease S' or set A to 1.
    - If $1/2 \leq w \leq \alpha$, then increase S'.
  - Subcase: $A = 1$
    - If $w \leq 1/4$, make no change in the state.
    - If $1/4 \leq w \leq 1/3$, then set A to 0, or set A to 0 and increase S', or decrease S', or make no change in the state.
    - If $1/3 \leq w \leq 1/2$, then set A to 0, or set A to 0 and increase S', or decrease S'.
    - If $1/2 \leq w \leq \alpha$, then increase S' or set A to 0.
  - Subcase: $B = 1$
    - If $w \leq 1/4$, make no change in the state.
    - If $1/4 \leq w \leq 1/3$, then set B to 0, or set B to 0 and increase S', or decrease S', or make no change in the state.
    - If $1/3 \leq w \leq 1 - \alpha$, then set B to 0, or set B to 0 and increase S', or decrease S'.
    - If $1 - \alpha \leq w \leq \alpha$, then set B to 0 or set B to 0 and increase S'.
- Case 2: $\frac{22}{100} \leq \frac{m}{k} \leq \frac{1}{4}$
  - Subcase: $A, B = 0$, $S' = 0$
    - If $w \leq 22/100$, make no change in the state.
    - If $22/100 \leq w \leq 1/4$, then set B to 1 or make no change in the state.
    - If $1/4 \leq w \leq 1 - \alpha$, then set B to 1.
    - If $1 - \alpha \leq w \leq 1/2$, then set A to 1.
    - If $1/2 \leq w \leq \alpha$, then increase S'.
  - Subcase: $A, B = 0$, $S' = 1$, and the remaining capacity of the bin corresponding to S' is less than $1 - \alpha$
    - If $w \leq 22/100$, make no change in the state.
    - If $22/100 \leq w \leq 1/4$, then set B to 1, or decrease S', or make no change in the state.
    - If $1/4 \leq w \leq 1 - \alpha$, then set B to 1 or decrease S'.

  * If $1 - \alpha \leq w \leq 1/2$, then set A to 1.
  * If $1/2 \leq w \leq \alpha$, then increase S'.

— Subcase: $A, B = 0, S' = 1$, and the remaining capacity of the bin corresponding to S' is less than $1 - \alpha$

  * If $w \leq 22/100$, make no change in the state.
  * If $22/100 \leq w \leq 1/4$, then set B to 1, or decrease S', or make no change in the state.
  * If $1/4 \leq w \leq 1 - \alpha$, then decrease S'.
  * If $1 - \alpha \leq w \leq 1/2$, then set A to 1 or decrease S'.
  * If $1/2 \leq w \leq \alpha$, then increase S'.

— Subcase: $A, B = 0, S' \geq 2$

  * If $w \leq 22/100$, make no change in the state.
  * If $22/100 \leq w \leq 1/4$, then set B to 1, or decrease S', or make no change in the state.
  * If $1/4 \leq w \leq 1 - \alpha$, then set B to 1 or decrease S'.
  * If $1 - \alpha \leq w \leq 1/2$, then set A to 1 or decrease S'.
  * If $1/2 \leq w \leq \alpha$, then increase S'.

— Subcase: $A = 1$

  * If $w \leq 22/100$, make no change in the state.
  * If $22/100 \leq w \leq 1/4$, then set A to 0, or set A to 0 and increase S', or decrease S', or make no change in the state.
  * If $1/4 \leq w \leq 1/2$, then set A to 0, or set A to 0 and increase S', or decrease S'.
  * If $1/2 \leq w \leq \alpha$, then set A to 0 or increase S'.

— Subcase: $B = 1$

  * If $w \leq 22/100$, make no change in the state.
  * If $22/100 \leq w \leq 1/4$, then set B to 0, or set B to 0 and increase S', or decrease S', or make no change in the state.
  * If $1/4 \leq w \leq 1 - \alpha$, then set B to 0, or set B to 0 and increase S', or decrease S', or make no change in the state.
  * If $1 - \alpha \leq w \leq \alpha$, then set B to 0, or set B to 0 and increase S'.

• Case 3: $\frac{12}{100} \leq \frac{m}{k} \leq \frac{22}{100}$

— Subcase: $A, B = 0, S' = 0$

  * If $w \leq 12/100$, make no change in the state.
  * If $12/100 \leq w \leq 22/100$, then set B to 1 or make no change in the state.
  * If $22/100 \leq w \leq 1 - \alpha$, then set B to 1.
  * If $1 - \alpha \leq w \leq 1/2$, then set A to 1.
  * If $1/2 \leq w \leq \alpha$, then increase S'.

— Subcase: $A, B = 0, S' = 1$, and the remaining capacity of the bin corresponding to S' is less than $1 - \alpha$

  * If $w \leq 12/100$, make no change in the state.
  * If $12/100 \leq w \leq 22/100$, then set B to 1, or decrease S', or make no change in the state.

* If $22/100 \leq w \leq 1 - \alpha$, then set B to 1 or decrease S'.
* If $1 - \alpha \leq w \leq 1/2$, then set A to 1.
* If $1/2 \leq w \leq \alpha$, then increase S'.

— Subcase: $A, B = 0, S' = 1$, and the remaining capacity of the bin corresponding to S' is at least $1 - \alpha$:

* If $w \leq 12/100$, make no change in the state.
* If $12/100 \leq w \leq 1 - \alpha$, then decrease S' or make no change in the state.
* If $1 - \alpha \leq w \leq 1/2$, then set A to 1, or decrease S', or make no change in the state.
* Note that in the last two cases above, the total range of w values that can cause S' to decrease is at most 22/100.
* If $1/2 \leq w \leq \alpha$, then increase S'.

— Subcase: $A, B = 0, S' \geq 2$:

* If $w \leq 12/100$, make no change in the state.
* If $12/100 \leq w \leq 1 - \alpha$, then set Z to 1, or decrease S', or make no change in the state.
* If $1 - \alpha \leq w \leq 1/2$, then set A to 1, or decrease S', or make no change in the state.
* If $1/2 \leq w \leq \alpha$, then increase S'.

— Subcase: $A = 1, S' = 0$:

* If $w \leq 12/100$, make no change in the state.
* If $12/100 \leq w \leq \alpha$, then note that in this case the total range of w values for which S' must increase and A must be set to 0 is at least $\alpha - 44/100$. In other cases, set A to 0 and increase S', or set A to 0, or increase S', or no change in the state.

— Subcase: $A = 1, S' = 1$, and the remaining capacity of the bin corresponding to S' is less than $1 - \alpha$:

* If $w \leq 12/100$, make no change in the state.
* If $12/100 \leq w \leq 22/100$, then set A to 0, or decrease S', or make no change in the state.
* If $22/100 \leq w \leq 1 - \alpha$, then set A to 0, set A to 0 and increase S', or decrease S'.
* If $1 - \alpha \leq w \leq 1/2$, then set A to 0, or set A to 0 and increase S'.
* If $1/2 \leq w \leq \alpha$, then increase S', or set A to 0, or set A to 0 and increase S'.

— Subcase: $A = 1, S' = 1$, and the remaining capacity of the bin corresponding to S' is at least $1 - \alpha$:

* If $w \leq 12/100$, make no change in the state.
* If $12/100 \leq w \leq 22/100$, then decrease S' or make no change in the state.
* If $22/100 \leq w \leq 1 - \alpha$, then decrease S'.
* If $1 - \alpha \leq w \leq 1/2$, then set A to 0, or set A to 0 and increase S', or decrease S', or make no change in the state.
* Note that in the last three cases above, the total range of w values that can cause S' to decrease is at most 22/100.
* If $1/2 \leq w \leq \alpha$, then increase S', or set A to 0, or set A to 0 and increase S'.

— Subcase: $A = 1$, $S' \geq 2$.

* If $w \leq 12/100$, make no change in the state.
* If $12/100 \leq w \leq 22/100$, then decrease S', or set A to 0 and increase S', or make no change in the state.
* If $22/100 \leq w \leq 1 - \alpha$, then decrease S', or set A to 0 and increase S', or set A to 0, or make no change in the state.
* If $1 - \alpha \leq w \leq 1/2$, then set A to 0, or set A to 0 and increase S', or decrease S', or make no change in the state.
* If $1/2 \leq w \leq \alpha$, then increase S', or set A to 0, or set A to 0 and increase S'.

— Subcase: $B = 1$, $S' = 0$

* If $w \leq 12/100$, make no change in the state.
* If $12/100 \leq w \leq \alpha$, then note that for $\alpha - 44/100$ of the w values in this range, S' must increase and B must be set to 0. In other cases, set B to 0 and increase S', or set B to 0 and set A to 1, or make no change in the state.

— Subcase: $B = 1$, $S' = 1$, and the remaining capacity of the bin corresponding to S' is less than $1 - \alpha$

* If $w \leq 12/100$, make no change in the state.
* If $12/100 \leq w \leq 22/100$, then set B to 0 and increase S', or set B to 0 and set A to 1, or decrease S', or make no change in the state.
* If $22/100 \leq w \leq 1 - \alpha$, then set B to 0 and increase S', or set B to 0 and set A to 1, or decrease S'.
* If $1 - \alpha \leq w \leq 1/2$, then set B to 0 and increase S', or set B to 0 and set A to 1, or set B to 0.
* If $1/2 \leq w \leq \alpha$, then set B to 0, or set B to 0 and increase S'.

— Subcase: $B = 1$, $S' = 1$, and the remaining capacity of the bin corresponding to S' is at least $1 - \alpha$

* If $w \leq 12/100$, make no change in the state.
* If $12/100 \leq w \leq 22/100$, then decrease S' or make no change in the state.
* If $22/100 \leq w \leq 1 - \alpha$, then decrease S'.
* If $1 - \alpha \leq w \leq 1/2$, then set B to 0 and increase S', or set B to 0 and set A to 1, or set B to 0, or decrease S'.
* Note that in the last three cases above, the total range of w values that can cause S' to decrease is at most 22/100.
* If $1/2 \leq w \leq \alpha$, then set B to 0, or set B to 0 and increase S'.

— Subcase: $B = 1$, $S' \geq 2$.

* If $w \leq 12/100$, make no change in the state.
* If $12/100 \leq w \leq 22/100$, then decrease S', or set B to 0 and increase S', or set B to 0 and A to 1, or make no change in the state.
* If $22/100 \leq w \leq 1 - \alpha$, then decrease S', or set B to 0 and increase S', or set B to 0 and set A to 1, or make no change in the state.
* If $1 - \alpha \leq w \leq 1/2$, then set B to 0 and increase S', or set B to 0 and set A to 1, or set B to 0, or decrease S'.
* If $1/2 \leq w \leq \alpha$, then set B to 0, or set B to 0 and increase S'.

• Case 4: $0 \leq \frac{m}{k} \leq \frac{12}{100}$

— Subcase: A, $B = 0$, $S' = 0$

* If $w \leq 12/100$, make no change or set B to 1.
* If $12/100 \leq w \leq 1 - \alpha$, then set B to 1.
* If $1 - \alpha \leq w \leq 1/2$, then set A to 1.
* If $1/2 \leq w \leq \alpha$, then increase S'.

— Subcase: A, $B = 0$, $S' = 1$, and the remaining capacity of the bin corresponding to S' is less than $\frac{24}{100}$

* If $w \leq 24/100$, then note the total range of w values that can cause S' to decrease is at most 12/100. Otherwise set B to 1 or make no change in the state.
* If $24/100 \leq w \leq 1 - \alpha$, then set B to 1.
* If $1 - \alpha \leq w \leq 1/2$, then set A to 1.
* If $1/2 \leq w \leq \alpha$, then increase S'.

— Subcase: A, $B = 0$, $S' = 1$, and the remaining capacity of the bin corresponding to S' is between $\frac{24}{100}$ and $1 - \alpha$

* If $w \leq 12/100$, then make no change in the state.
* If $12/100 \leq w \leq 1 - \alpha$, then note the total range of w values that can cause S' to decrease is at most 12/100. Otherwise set B to 1 or make no change in the state.
* If $1 - \alpha \leq w \leq 1/2$, then set A to 1.
* If $1/2 \leq w \leq \alpha$, then increase S'.

— Subcase: A, $B = 0$, $S' = 1$, and the remaining capacity of the bin corresponding to S' is at least $1 - \alpha$

* If $w \leq 88/100 - \alpha$, then make no change in the state.
* If $88/100 - \alpha \leq w \leq 1/2$, then note the total range of w values that can cause S' to decrease is at most 12/100. Otherwise set A to 1 or make no change in the state.
* If $1/2 \leq w \leq \alpha$, then increase S'.

— Subcase: A, $B = 0$, $S' = 2$, and the largest remaining capacity of a bin corresponding to S' is less than $\frac{24}{100}$

* If $w \leq 24/100$, then decrease S', set B to 1, or make no change in the state.
* If $24/100 \leq w \leq 1 - \alpha$, then set B to 1.
* If $1 - \alpha \leq w \leq 1/2$, then set A to 1.
* If $1/2 \leq w \leq \alpha$, then increase S'.

— Subcase: A, $B = 0$, $S' = 2$, and the largest remaining capacity of a corresponding to S' is between $\frac{24}{100}$ and $1 - \alpha$

* If $w \leq 1 - \alpha$, then note the total range of w values that can cause S' to decrease is at most 24/100. Otherwise set B to 1 or make no change in the state.
* If $1 - \alpha \leq w \leq 1/2$, then set A to 1.
* If $1/2 \leq w \leq \alpha$, then increase S'.

— Subcase: A, $B = 0$, $S' = 2$, and the largest remaining capacity of a bin corresponding to S' is between $1 - \alpha$ and $\frac{36}{100}$

* If $w \leq 36/100$, then note the total range of w values that can cause S' to decrease is at most 24/100. Otherwise set A to 1 or make no change in the state.

  * If $36/100 \leq w \leq 1/2$, then set A to 1.
  * If $1/2 \leq w \leq \alpha$, then increase S'.

— Subcase: A, $B = 0$, $S' = 2$, and the largest remaining capacity of a bin corresponding to S' is at least $\frac{36}{100}$

  * If $w \leq 36/100$, then note the total range of w values that can cause S' to decrease is at most 24/100. Otherwise set A to 1 or make no change in the state.
  * If $36/100 \leq w \leq 1/2$, then set A to 1 or decrease S' or make no change in the state.
  * If $1/2 \leq w \leq \alpha$, then increase S'.

— Subcase: A, $B = 0$, $S' \geq 3$, and the largest remaining capacity of a bin corresponding to S' is less than $1 - \alpha$

  * If $w \leq 1 - \alpha$, then decrease S', set B to 1, or make no change in the state.
  * If $1 - \alpha \leq w \leq 1/2$, then set A to 1.
  * If $1/2 \leq w \leq \alpha$, then increase S'.

— Subcase: A, $B = 0$, $S' \geq 3$, and the largest remaining capacity of a bin corresponding to S' is at least $1 - \alpha$

  * If $w \leq 1/2$, then note the total range of w values that can cause S to decrease is at most $12 \cdot S'/100$. Otherwise set A to 1 or make no change in the state.
  * If $1/2 \leq w \leq \alpha$, then increase S'.

— Subcase: $A = 1$, $S' = 0$

  * Note the total range of w values that cause S' to increase (possibly but not necessarily setting A to 0) is 38/100. Otherwise increase S', or increase S' and set A to 0, or set A to 0, or make no change in the state.

— Subcase: $A = 1$, $S' \geq 1$

  * Note the total range of w values that cause S' to decrease is at most $\min(12 \cdot S'/100, 1/2)$. Otherwise increase S', or increase S' and set A to 0, or set A to 0, or make no change in the state.

— Subcase: $B = 1$, $S' = 0$, and the bin corresponding to B has remaining capacity less than 4/5

  * Note the total range of w values that cause S' to increase and B to be set to 0 is at least $\alpha - 3/10$.
  * Note the total range of w values that allow no change in the state is at most $4/5 - \alpha$.
  * Otherwise increase S' and set B to 0, set A to 1 and set B to 0, set B to 0.

— Subcase: $B = 1$, $S' = 0$, and the bin corresponding to B has remaining capacity at least 4/5

  * Note the total range of w values that cause S' to increase and B to be set to 0 is at least $\alpha - 1/2$.
  * Note the total range of w values that cause A to be set to 1 and B to be set to 0 is at least $\alpha - 1/2$.
  * Otherwise increase S' and set B to 0, set A to 1 and set B to 0, or make no change in the state.

— Subcase: $B = 1$, $S' \geq 1$

* Note the total range of w values that cause S' to decrease is at most $\min(12 \cdot S'/100,\ 1/2)$.
* Note the total range of w values that cause S' to increase and B to be set to 0 is at least $\alpha - 62/100$.
* Otherwise increase S' and set B to 0, set A to 1 and set B to 0, set B to 0, or make no change in the state.

ACKNOWLEDGMENTS

The authors especially wish to thank David Johnson for his helpful suggestions. The authors also extend a great deal of thanks to one of the anonymous referees for their diligence beyond the call of duty in finding errors in the original text and providing a great deal of advice on how to present the arguments more clearly.

REFERENCES

[1] Problems in analysis of algorithms pages, http://pauillac.inria.fr/algo/AofA/Problems/index.html. Problem 5, posted by Ed Coffman.

[2] S. Albers and M. Mitzenmacher, Average case analyses of First Fit and Random Fit bin packing, Proc 9th Annu ACM Symp Discrete Alg, 1998, pp. 290–299.

[3] D. P. Bertsekas and J. N. Tsitsiklis, Neuro-dynamic programming, Athena Scientific, Boston, 1996.

[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, Proc 23rd Annu ACM Symp Theory Comput, 1991, pp. 230–240.

[5] E. G. Coffman, Jr., M. R. Garey, and D. S. Johnson, Approximation algorithms for bin packing: A survey, Approximation algorithms for NP-hard problems, D. S. Hochbaum (Editor), PWS, Boston, 1996.

[6] E. G. Coffman, Jr., D. S. Johnson, P. W. Shor, and R. R. Weber, Markov chains, computer proofs and average-case analysis of Best Fit bin packing, Proc 25th Annu ACM Symp Theory Comput 1993, pp. 412–421.

[7] M. Hofri, Analysis of algorithms—mathematical methods, computational tools, Oxford University Press, Oxford, October 1995.

[8] 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.

[9] N. Karmakar, Probabilistic analysis of some bin-packing problems, IEEE FOCS 1982, pp. 107–111.

[10] C. Kenyon, Y. Rabani, and A. Sinclair, Biased random walks, Lyapunov functions, and stochastic analysis of Best Fit bin packing, J Alg 27(2) (1998), 218–235.

[11] F. T. Leighton and P. W. Shor, Tight bounds for minimax grid matching with application to the average case analysis of algorithms, Combinatorica 9 (1989), 161–187.

[12] J. Mosely and P. A. Humblet, A class of efficient contention resolution algorithms for multiple access channels, IEEE Trans Commun 33(2) (1985), 145–151.

[13] M. Puterman, Markov decision processes: Discrete stochastic dynamic programming, Wiley, New York, 1994.

[14] M. Shaked and J. G. Shanthikumar, Stochastic orders and their applications, Academic Press, San Diego, 1994.

[15] P. W. Shor, The average case analysis of some on-line algorithms for bin packing, Combinatorica 6 (1986), 179–200.