Divergence-proving Techniques for Best Fit Bin Packing and Random Fit

Senior Thesis
presented by
Petar Maymounkov
to **Mathematics**
in partial fulfillment of the honors requirements
for the degree of
Bachelor of Arts
Harvard College
Cambridge, Massachusetts

April 2, 2001

Abstract

This work discusses my attempts to extend Kenyon and Mitzenmacher's technique for proving diveregnce of the online approximation algorithm Best Fit to Random Fit – another approximation algorithm for the well-known NP-hard problem of bin packing. In specific, the paper goes over Kenyon and Mitzenmacher's recent advances on divergence of the waste of Best Fit bin packing for the skewed distributions $U\{\alpha k, k\}$ with $\alpha \in [0.66, 2/3)$ in detail, and describes the modifications I made to their methods in attempt to prove diveregence for Random Fit under the same input conditions.

Acknowledgements

I would like to thank my thesis adviser Professor Michael Mitzenmacher for waking my appreciation for the problems of Computer Science, and my interest in stochastic processes and their applications to algorithms. I am very glad that he had the enthusiasm to share his views of Computer Science in his great research classes, because they have made me decide to continue my career in that field.

Thanks to Professor Leslie Valiant and to Professor Michael Rabin for the excellent lectures and advice that they have given throughout my years at Harvard.

Thanks to Christian Crudder for proof-reading my thesis and providing valuable suggestions.

1 Introduction

Bin packing is a classical NP-hard problem in Computer Science. Its definition is fairly simple, yet its behavior appears to be rather complex. In the one-dimensional version one is given a finite set of items of sizes $L_n = a_1, \ldots, a_n \in (0, 1]$ and is asked to find a packing of these items into bins of capacity 1, so that the number of bins used is minimized. Since this is an NP-hard problem, people have turned their attention to analysing different approximation techniques. A rich spectrum of algorithms for approximating the bin packing problem has been looked at, yet the most prominent seem to be the two *on-line* algorithms *Best Fit* and *First Fit*. In the on-line setting, one receives an uninterrupted sequence of items and is asked to pack them into bins as they come. The *Best Fit* algorithm places each successive item in the bin with the smallest residual capacity that can accomodate the item; if no such bin exists, a new bin is created and the item is placed there. In the *First Fit* algorithm, one keeps track of the order in which bins are created $b_1, b_2, \ldots$ and places each successive item in the bin with the smallest index that can accommodate the item, or otherwise creates a new one.

In 1974, Johnson *et al.* [6] showed that the worst-case performance of both Best and First Fit is within a factor of 1.7 of the optimal packing. However, the worst-case input sequences seemed to be too rare combinatorically, which motivated the study of all sorts of of bin packing algorithms under the so-called skewed distribution $U(0, a)$, where the items' sizes are independent random variables, uniform over the interval $[0, a]$, for $a < 1$. For these distributions, the optimal packing was proved to be *perfect* [5], in the sense that $\lim_{n\to\infty} E\big[OPT(L_n)/(a_1 + a_2 + \cdots + a_n)\big] = 1$ [1], which means that the *expected* asymptotic performance ratio of any on-line algorithm is strictly greater than 1 if and only if the *waste*[2] grows linearly in the number of items. Based on experimental observation it was conjectured that for all skewed distributions $U(0, a)$ the waste grows linearly for both Best [1, 9] and First Fit [4].

Later on, Coffman *et al.* [7] introduced the discrete skewed distributions $U\{j, k\}$ as means of gaining insight into the continuous case. Under the distribution $U\{j, k\}$ the items' sizes are drawn independently and uniformly from the set $\{1/k, 2/k, \ldots, j/k\}$. $U\{j, k\}$ approximates $U(0, a)$, if we set $j = ak$ and let k grow to infinity. In addition, an easier way to think about $U\{j, k\}$, is to define the bins to be of capacity k and the items' sizes be uniformly distributed over $\{1, 2, \ldots, j\}$. In 1995 Kenyon, Rabani and Sinclair [3] proved that Best Fit has $O(1)$ waste under $U\{k-2, k\}$. Then in 1997 Albers and Mitzenmacher [2] used a new algorithm called *Random Fit* (RF) to make a transition from Best to First Fit and prove that First Fit also has $O(1)$ waste under $U\{k-2, k\}$. In this respect Random Fit is interesting as a bridge between First and Best. As in First Fit, Random Fit keeps track of the order in which bins are created $b_1, b_2, \ldots$ and when a new item comes, the sequence of bins is permuted randomly and uniformly, and the item is packed in the bin with the lowest index that can accommodate it; otherwise a new bin is created and the item is placed there. Another way of describing Random Fit is to say that when a new item comes, Random Fit finds the subset of bins which can accommodate it and picks a random one from it to place the item there.

Aside from the two results concerning waste performance mentioned above, there has been very little proven. However, it is conjectured that both Best and First Fit have linear waste for almost all skewed distributions $U\{j, k\}$. Very recently, Kenyon and Mitzenmacher [4] proved

[1] Here $OPT(L_n)$ is the optimal number of bins (respectively their aggregate capacity) that can pack the items from the sequence L_n.

[2] *waste* refers to the difference between the total capacity of the bins used and the aggregate size of the items packed.

linear waste of Best Fit for $U\{j,k\}$ where $j = \alpha k$ and $0.66 \leqq \alpha < 2/3$. The proof of the result is based on a very specific property that the algorithm exhibits when j is within this small range. The present paper goes in detail over their paper and fills in certain lemmas and discussions which have been omitted for bravity; then I discusses why it is hard to apply their methodology to Random Fit.

2 Sketch of Ideas

The now standard way of thinking about bin packing under the discrete distribution $U\{j,k\}$ is a Markov chain, introduced in [8]. At any moment of time we can have bins with remaining capacities in the set $\{1, 2, \ldots, k-1\}$; we let s_i to be the number of bins with remaining capacity i, and treat the vector $s = (s_1, s_2, \ldots, s_{k-1})$ as our Markov state. Under Best and Random Fit it is clear how the states are connected, since the vector s carries all the information necessary to compute the probabilities of next moves. This makes it a multi-dimensional $\mathbb{Z}_+^{k-1}$ Markov chain.

As mentioned in the introduction, to prove divergence from optimal, we need to show that the waste grows linearly with time. This is equivalent to showing that asymptotically the Markov chain process goes away from the origin. The approach that [4] takes, and which I am going to follow, is to look a different Markov chain, and consequently show that s_1 grows linearly. Deriving the new Markov chain from the one mentioned above is a bit cumbersome, but there is a shorter and more intuitive way of describing it. One can divide the possible configuratons [3] of the process into two groups *easy* and *difficult*; when in an easy configuration the probability of increasing s_1 will be easily shown to be more than 1/2; in a difficult configuration the probability of increasing s_1 will be less than 1/2. A novel lemma due to [4] shows that when $j < 2/3k$, the difficult configurations are very few. The strategy is to prove that difficult configurations are short-lived, in the sense that their effect on s_1 is amortized if we run the process for a constant number C of additional steps. The state of the algorithm when it enters a difficult configuration, together with the following C steps become a single *superstep*, represented by a single node in the new Markov chain. Next we are going to show that most transitions in the new Markov chain increase s_1. This entails showing that transitions going out of supersteps tend to increase s_1.

It is hard to analyse j^C possibilities (the possible outcomes of a superstep), so Kenyon and Mitzenmacher propose to classify all configurations into finitely many groups and analyse the worst-case behavior based only on group information by using stochastic domination. The worst-case outcome is determined by dynamic programming: one should succesively find the worst outcome for each group as a starting point with remaining 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. Dynamic programming is commonly used in Markov decision processes (see e.g. [10]). Yet [4] is the first time when it is used in the contex of average-case bin packing analysis. Although the derived dynamic program has to deal with a constant number of cases, there are tens of thousands so a computer program is necessary.

3 Analysis of discrete skewed distributions of Best Fit

This section proves the divergence of waste for Best Fit. The proof is a more detailed version of [4]'s proof.

[3] By *configuration*, I mean a vector $s = (s_1, s_2, \ldots, s_{k-1})$.

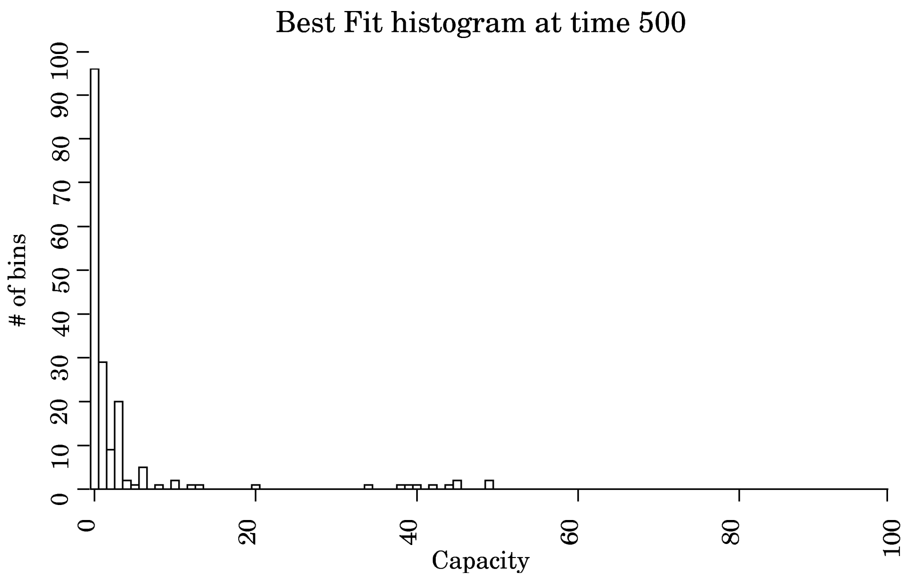


Figure 1: The histogram above represents a convenient way of thinking about the state of the Best Fit bin packing process. On the X-axis are all possible bin capacities, and the Y-axis shows the number of bins of each given capacity. If one were to observe this histogram in the actual process of bin packing, one would roughly see new tics appearing in the right part of the X-axis, travelling in a jumpy fashion towards the left piles.

3.1 The general Markov chain

Throughout this section, I am going to be working with the discrete distribution $U\{j, k\}$, where the bins are of capacity k and the incoming items are uniformly distributed over $\{1, 2, \ldots, j\}$.

Let's reiterate the standard Markov chain definition from [8]. Denote by $s_i(t)$ the number of bins of residual capacity exactly i at time t. Call $s(t) = (s_1(t), s_2(t), \ldots, s_{k-1}(t))$ the Markov state or configuration vector. The process starts at $s(0) = (0, 0, \ldots, 0)$, reflecting the fact that there are no open bins. Let l be the size of the next item inserted. Let i be the smallest index such that $i \geqq l$, if such exists; in that case, Best Fit inserts item l into a bin with capacity i, so we have $s_i(t+1) = s_i(t) - 1$ and, if $i > l$, $s_{i-l}(t+1) = s_{i-l}(t) + 1$; all other components of $s(t)$ remain unchanged. If no such sequence exists, then the algorithm inserts item l into an empty bin, so we have $s_{k-l}(t+1) = s_{k-l}(t) + 1$ and all other components of $s(t)$ remain unchanged. This completes the description of the Markov chain (it gives enough information to derive the capacities on the Markov chain's edges).

From now on, it is assumed that $j = \alpha k$, where $0.66 \leqq \alpha \leqq 2/3$. The remainder of the section builds a different Markov chain for analysing the packing process, but uses the language from the definition of this general chain for convenience. Also, in the paper we assume that k is sufficiently large, so that all arguments hold throughout.

3.2 The difficult configurations

The attack for proving instability is to show that s_1, the number of almost full bins, is biased upward and hence tends to increase. Let $X_t \in \{0, 1, \ldots, j-1\}$ denote the number of ways to increase s_1, and $Y_t \in \{0, 1\}$ denote the number of ways to decrease s_1 at time t.

In the case of Best Fit, the values of s_1 increase exactly when an item of size x is inserted and we have $s_x = 0$ and $s_{x+1} \neq 0$, and so X_t is exactly the number of such pairs (s_x, s_{x+1}) for $0 < x \leqq j$. At every time step, if $s_1(t) = 0$, we have $Y_t = 0$, and if $s_1(t) \neq 0$, $Y_t = 1$: namely $s_1(t)$ can decrease only when an item of size 1 arrives. The only situations when s_1 is biased downward are when s_1 has no way to increase and one way to decrease, i.e. for some $m \geqq 1$, $s_1, s_2, \ldots, s_m \neq 0$ and $s_{m+1} = \cdots = s_{j+1} = 0$. We call these configurations when s_1 is biased downward *difficult configurations*, as handling them is the challenge of the problem. The so-called open range lemma in the next section shows that m must be less than $k/3$.

3.3 The open range lemma

The following lemma demonstrates that one cannot have more than one bin with remaining capacity within a rather large range. The following fact is a classical basic property of both Best, First and Random Fit.

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

Moreover, for the sake of completeness we mention one more fact, which follows from Fact 1.

Fact 2. *At any time of the execution of Best, First or Random Fit the following inequality holds:*

$$\sum_{\lceil k/2 \rceil \leqq i \leqq k-1} s_i \leqq 1.$$

Lemma 1 (Open range lemma). *If the maximum item size is j, and $j = \alpha k$ with $\alpha < 2/3$, then $s_{k/3} + \cdots + s_{k-j-1} \leqq 1$.*

Proof. Assume $3 \| k$ for clarity. Since $s_{k/3} + \cdots + s_{k-j-1} = 0$ initially, we only need to 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 $k/3 \leqq i' \leqq k - j - 1$. How can $s_{i'}$ increase? 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 j, the largest item size. Thus a bin with remaining capacity i' can only be introduced by adding some item of size x to a bin which already has a remaining capacity $g > k-j-1$, with $g - x = i'$.

Assume then that at time t there is one bin with remaining capacity i and one with remaining capacity g. From Fact 1, we have $g + i < k$, so that

$$k - g > i \geqq \frac{k}{3}. \tag{1}$$

By definition, the remaining capacity g must be larger than i. Also, by the definition of Best Fit, item x would have been placed in the bin with remaining capacity i if it had fit there, rather than in the bin with remaining capacity g. So it must be that x does not fit in remaining capacity i:

$$x > i \geqq \frac{k}{3}. \tag{2}$$

Now, by assumption

$$i' \geqq \frac{k}{3}. \tag{3}$$

Summing inequalities (1),(2) and (3) we obtain $k - g + x + i' > k$, and hence $i' > g - x$ – a contradiction. □

Again, 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 any difficult configuration must have $m \leqq k/3$. We note that this result holds for Random Fit as well, since both Fact 1 and 2 hold for the two algorithms.

3.4 A Markov chain of supersteps

Recall that difficult configurations are the ones such that for some $m \leqq k/3$, $s_1, s_2, \ldots, s_m \neq 0$ and $s_{m+1} = \cdots = s_{j+1} = 0$. When the process enters a difficult configuration, we consider the evolution of the system over τ steps for some random *stopping time* τ (defined precisely in the next paragraph), bounded by a constant C independent of k or j. We show that the probability of s_1 to increase after these τ steps is more than 1/2. In other words, we are creating a Markov chain which starts at the initial configuration, branches out and every time it reaches a difficult configuration it is collapsed together with the following τ steps in one Markov state, this state we call a *superstep*. As we show in what follows, there are lots of transitions going out of superstep states, in the order of j^C, but for the majority of them s_1 had increased on exit of the superstep. All in all, we end up with a Markov chain, for which the sum of the probabilities of the outgoing edges of each state which lead to an increase of s_1 is more than the sum of the probabilities of the outgoing edges which lead to a decrease in s_1. It is a theorem, which we give at the end, that states that from this condition for the Markov chain it follows that s_1 is divergent.

Now, let's turn to defining the stopping time framework more precisely. Suppose we have just entered a difficult configuration, and assume for simplicity that we are at time 0 now. Then the stopping time τ will correspond to one of the following events:

1. Time step C has been reached for some fixed constant C.
2. The coordinate s_1 increases.
3. The coordinate s_1 decreases.
4. The coordinate s_m becomes 0.
5. The coordinate s_{m+1} becomes positive.
6. For some $m + 1 < a$, the coordinate s_a becomes equal to 2. In fact, $a < \lceil k/2 \rceil$, due to Fact 2.
7. For some $m + 2 < a$, the coordinates s_a and s_{a-1} become positive. In fact, $a \leqq \lceil k/2 \rceil$, again due to Fact 2.

The idea of this stopping time is to show that when we run the process for at most C more steps, after a difficult configuration has occured, it is more likely that s_1 will either increase or remain unchanged than s_1 will decrease. Events 4,5,6 and 7 are there, because they make the analysis simpler. They are very unlikely as it turns out. As a result, s_1 is either unbiased or biased upwards over any normal step or any superstep of the chain, and this is sufficient to prove instability.

In the analysis of a superstep (the evolution of the process from entering into a difficult configuration until the stopping time), we shall assume that at each step the number of ways in which s_1 can increase is $s_{m+2} + s_{m+3} + \cdots + s_{j+1}$. This assumption will make the furthe analysis much simpler, but it fails to be true when either $s_a > 1$ for $m + 2 \leqq a \leqq j + 1$, or $s_{b-1}, s_b > 0$ for some $m + 2 < b \leqq j + 1$; it also fails to be true when for some $0 \leqq a < m$ s_a becomes

0, but we shall disregard this case, because then the actual possibilities for increasing s will be even more; so, that is why we introduce the stopping events 4,5,6 and 7, which make sure that $s_{m+2} + s_{m+3} + \cdots + s_{j+1}$ is truely equal (or smaller than) the number of ways s_1 can increase at any time throughout the superstep. In plain words, every bin whose residual capacity turns to a for $m + 2 \leqq a \leqq j + 1$ is useful unless $s_a > 0$ or $s_{a-1} > 0$.

Now, turning to the analysis more rigorously, we wish to show that the probability that s_1 has increased after the end of the τ-th step is greater than the probability that it has decreased. Imagine that we run the process for C steps regardless of what happens, but we put a mark at the τ-th time step. So we actually wish to show that the probability that s_1 has increased before we put the mark is greater than the probability that it has decreased.

Instead of examing all j^C possible outcomes, we introduce another simplification, we are only going to look at all possible sequences $X_1, \ldots, X_{C-1}$, where X_t will be a random variable,representing the number of ways for s_1 to increase at time t for $t \in [0, C)$. Denote by $A(t)$, for $0 \leqq t < C-1$, the probability that time t is not the stopping time, i.e. $t \neq \tau$, on the condition that time t has already been reached. Taking into account that $X_t \leqq C$ we get:

$$\begin{aligned} A(t) &\geqq 1 - \left[\frac{X_t}{j} + \frac{1}{j} + \frac{1}{j} + \frac{X_t + 2}{j} + \frac{X_t^2}{j} + \frac{X_t^2}{j}\right] && (4) \\ &\geqq 1 - \left[\frac{C}{j} + \frac{1}{j} + \frac{C+2}{j} + \frac{C^2}{j} + \frac{C^2}{j}\right] && (5) \\ &= 1 - \frac{C^2 + 2C + 3}{j}, && (6) \end{aligned}$$

where the terms in the square brackets of the first line above are upper bounds on the probabilties of the events 2,3,4,5,6 and 7. Let's look at the probability $\overline{P}$ that s_1 has increased after the τ-th step.

$$\begin{aligned} \overline{P} &= \sum_{t=1}^{C} \left[\frac{X_t}{j} \prod_{i=1}^{t-1} A(i)\right] && (7) \\ &\geqq \frac{1}{j} \sum_{t=1}^{C} X_t + O\left(\frac{1}{j^2}\right) && (8) \end{aligned}$$

Similarly, an upper bound on the probability $\underline{P}$ that s_1 has decreased after the τ-th step would be

$$\underline{P} \leqq \frac{C}{j}. \tag{9}$$

From the above two inequalities we get a lower bound on the difference between the probability that s_1 increases and the probability that s_1 decreses.

$$\overline{P} - \underline{P} \geqq \frac{\sum_{i=1}^{C} X_i - C}{j} + O\left(\frac{1}{j^2}\right). \tag{10}$$

Let $\overline{P}(x_1, \ldots, x_C)$ be the probability that s_1 increases, given that $X_i = x_i$, for $1 \leqq i \leqq C$; let

$P_{x_1,\ldots,x_C}$ be the probability that $X_i = x_i$. Then we obtain a closed form lower bound for $\overline{P}$.

$$\overline{P} \geqq \sum_{x_1,\ldots,x_C} \overline{P}(x_1,\ldots,x_C)P_{x_1,\ldots,x_C} \tag{11}$$

$$= \sum_{x_1,\ldots,x_C} \left[\frac{1}{j}\sum_{i=1}^{C} x_i + O\left(\frac{1}{j^2}\right)\right] P_{x_1,\ldots,x_C} \tag{12}$$

$$= \frac{1}{j} E\left[\sum_{i=1}^{C} X_i\right] + O\left(\frac{1}{j^2}\right) \tag{13}$$

So, for $\overline{P} - \underline{P}$ we need to show:

$$\overline{P} - \underline{P} = \frac{C}{j}\left[\frac{E\left[\sum_{i=1}^{C} X_i\right]}{C} - 1\right] + O\left(\frac{1}{j^2}\right) \tag{14}$$

$$\geqq 0. \tag{15}$$

Instead, we are going to show that

$$\frac{C}{j}\left[\frac{E\left[\sum_{i=1}^{C} X_i\right]}{C} - 1\right] > \varepsilon, \tag{16}$$

for some $\varepsilon > 0$. This would be enough, because as k grows to infinity, so does j and the $O(1/j^2)$ term in $\overline{P} - \underline{P}$ will become smaller than ε in absolute value.

Intuitively, (16) shows that we can count the number of ways s_1 can increase at each step and subtract the number of ways s_1 can decrease at each step over C steps, in order to compute difference in the probability that s_1 increases rather than decreases over an interval that ends at a stopping time.

As we said before, from the definition of the stopping time, which excludes problematic events, it follows that X_t, the number of ways for s_1 to increase, is equal to (or greater than) $s_{m+2} + \cdots + s_{j+1}$ at each time t. Thus we are reduced to showing that there exists a constant C, a real $\varepsilon' > 0$ (both independent of k) such that for sufficiently large k,

$$E\left[\frac{\sum_{t=1}^{C}\left(s_{m+2}(t) + s_{m+3}(t) + \cdots + s_{j+1}(t)\right)}{C} \,\middle|\, \text{configuration at time } 0\right] > 1 + \varepsilon'. \tag{17}$$

Moreover, the formula above says that we are going to disregard the cases when a superstep stops prematurely (before C steps have past), and consider only the remaining ones.

3.5 Tha analysis starting from a difficult configuration

In this subsection we need the assumtion that $j = \alpha k$ for $\alpha \in (33/50, 2/3)$, and using the fact that $m \leqq k/3$ we analyze the life of a superstep, starting at a difficult configuration with $s_1, \ldots, s_m \neq 0$ and $s_{m+1} = \cdots = s_{j+1} = 0$.

3.5.1 Using stochastic domination: a nonrigorous example

First, let's look at a simplified example, which demonstrates the gist of the analysis of the (general) Markov chain within a superstep. For this simplified analysis we shall ignore the effect of non-empty bins with remaining capacity at least $k/2$ (in fact, there can be at most one such bin due to Fact 2), i.e. we assume $s_{k/2} + \cdots + s_{k-1} = 0$. As explained later, such mostly empty bins complicate the analysis.

We start with an example where $k/4 \leqq m \leqq k/3$. At every step there are $j = \alpha k$ possibilities for the item arriving. Now we want to calculate a lower bound in the probability of increasing and an upper bound on the probability of decreasing s_1. At this point we should consider the conditional probabilities of X_t's change, on the condition that we are not entering a stopping configuration. But since the ways in which we can enter a stopping configuration are constantly many $O(C)$, as k grows to infinity, their impact becomes negligible. Hence, we can safely assume that no incoming will item will throw us in a stopping configuration. So, out of the $j = \alpha k$ possibilities, X_t has at least $j - k/2 = k(2\alpha - 1)/2$ ways of increasing, corresponding to insertions of items $k/2, k/2+1, \ldots, j$ (this is due to the assumption that bins are never more than half empty). On the other hand, since s_m is positive, X_t has at most $k/2 - m \leqq k/6$ ways of decreasing if it is non-zero, and no ways of decreasing of it is equal to 0. It is also worth noting that in general X_t has at most mX_t ways of decreasing in general, since for a bin that contributes to X_t, there could possibly be at most m items that on entrance could reduce its residual capacity to something not greater than m. (For this range of m, however, the bound of $k/2 - m$ is better.)

We now use stoachstic domination to justify our analysis, given a lower and an upper bound on the probabilities of X_t increasing and decreasing, respectively. Following standard definitions (see, e.g. [11]), we say that X stochastically dominates Y and write $X \geqq_R Y$, if $\Pr[X > u] \geqq \Pr[Y > u]$ for all real values u. It follows from that very definition that $E[X] \geqq E[Y]$, and intuitively X is more likely to take on larger values than Y. It is simple to show now by induction on t that $X_t \geqq_R Z_t$, where

$$Z_0 = 0, \tag{18}$$

$$Z_{t+1} = \begin{cases} Z_t + 1, & \text{w.p. } \frac{k(2\alpha-1)}{2k\alpha}, \\ Z_t - 1, & \text{w.p. } \frac{k/2-m}{k\alpha}, \text{ if } Z_t \neq 0, \text{ 0 otherwise,} \\ 0, & \text{with all remaining probability.} \end{cases} \tag{19}$$

It follows from the definition above that the probability of Z_t increasing is smallest, and the probability of Z_t decreasing is largest for the smallest value of α, $\alpha = 33/50$, and the smallest value of m, $m = k/4$. That is, $X_t \geqq_r Z_t$, where

$$Z_0 = 0, \tag{20}$$

$$Z_{t+1} = \begin{cases} Z_t + 1, & \text{w.p. } 8/33, \\ Z_t - 1, & \text{w.p. } 25/66, \text{ if } Z_t \neq 0, \text{ 0 otherwise,} \\ 0, & \text{with all remaining probability.} \end{cases} \tag{21}$$

By explicitly computing the distributions of $Z_0, Z_1, \ldots, Z_{C-1}$, we find that $E\big[(Z_0 + Z_1 + \cdots + Z_{C-1})/C\big] > 1$ for a small constant C. (For $C = 30$, $E\big[(Z_0 + Z_1 + \cdots + Z_{C-1})/C\big] \approx 1.007$; as C gets large this value increases.) From here, since X dominates Z stochastically, we get, for a small constant C, that

$$E\big[(X_0 + X_1 + \cdots + X_{C-1})/C\big] \geqq E\big[(Z_0 + Z_1 + \cdots + Z_{C-1})/C\big] > 1. \tag{22}$$

We can continue by breaking down the process into a finite number of similar cases, covering the entire range of m. (For example, we could take the cases $0 \leqq m \leqq k/8$ and $k/8 \leqq m \leqq k/4$.) In

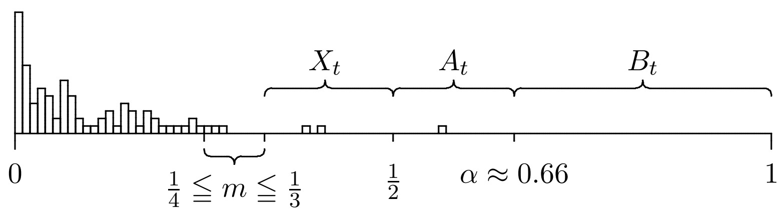


Figure 2: Regions of the Best Fit histogram, represented by the configuration (X_t, A_t, B_t). The range for m represents the fact that in the actual case analysis (given in the appendix) m is assumed to be within some range, but it is not clear what it's exact value is.

each case, we can find the probability of X_t increasing or decreasing, so as to find a dominating simple one-dimensional random walk. It is thus easy to check each specific case, simply by determining the distribution of Z_t over a reasonably small number of steps.

3.5.2 The important parameters describing a configuration

The above example demonstrates the general approach: instead of looking explicitly at all possible configurations into which the process can go, we group them together by the number of ways they present for s_1 to increase, and examine the evolution of the system for a small number of steps. In this example, however, we ignored the impact of "mostly empty" bins (the ones with $s_i > 0$ and $i \geqq k/2$) on the outcome of the process, but in fact it is quite important. Take for example the case when $j = 0.34k$, $m = 0.32k$ and $s_{0.64k} = 1$. In this case it is impossible for any of $s_{m+2}, s_{m+3}, \ldots, s_{k/2}$ to increase on the next step. This example is meant to show that the presence of mostly empty bins has an impact on the process.

In order to account for these mostly empty bins, we refine the grouping of the configurations to account for them too. Call a bin with remaining capacity at least $k/2$ a *light bin.* According to Fact 2 there can only be one such bin at a time. In fact, we refine the grouping further by distinguishing between two types of light bins. Call a light bin *helpful* if its remaining capacity is at most $j+1$. A helpful light bin can immediately lead to an increase in s_1 (given that we are within a superstep), if an appropriately sized item arrives. Similarly, call a bin *unhelpful* if its remaining capacity is greater than $j+1$.

Now we can subdivide all possible configurations into groups of the form (X_t, A_t, B_t), where X_t is again the number of ways for s_1 to increase at time 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 light unhelpful bin. Note that when $A_t = 1$, we must have $X_t \geqq 1$, since by definition a helpful light bin provides one way for s_1 to increase.

An illustration of a configuration is given on Figure 2.

3.6 The configuration space

Before jumping into the specific configuration framework that I have picked for Random Fit, I'd like to say a few words about configuration spaces in general.

3.6.1 The dynamic program

Our approach for showing that $E\big[\sum_{t=1}^{C}\big(s_{m+2}(t)+s_{m+3}(t)+\cdots+s_{j+1}(t)\big)/C \,\big|\, \text{config. at time } 0\big]> 1+\varepsilon'$ would be as follows. Start with the initial difficult configuration on entry to the superstep $(0,0,0)$. Then compute the distribution over all possible configurations after the first step. Then we use this distribution to compute the distribution of the configurations after the second step, and so forth. At each step we compute $E\big[\sum_{t=1}^{C}\big(s_{m+2}(t)+s_{m+3}(t)+\cdots+s_{j+1}(t)\big)/C \,\big|\, \text{config. at time } 0\big]$, and stop as soon as it is greater than 1 by a reasonable fraction (a couple of orders of magnitute bigger than the floating-point precision of the 32-bit PC processor, which we used to analyse the outcomes).

Unfortunately, this relatively simple way of attacking the problem doesn't work, due to the fact that our triple (X_t, A_t, B_t) does not carry complete information about the current configuration, and so for certain sizes of the incoming items there will be ambiguity in deciding what the next configuration would be. We want to use worst-case analysis in such situations, similar to allowing an oblivious adversary some limited power in deciding the flow of the process. For example, suppose $A_t = 1$, and an item of size in the range $[0.5k, j]$ arrives. Such an item could be placed in the light bin; alternatively, such an item may be too large for the light bin, and instead cause x_t to increase. The effect of the item depends on the exact residual capacity of the light bin and the value of the entering item; however, we don't keep track of the capacity of the light bin in our state.

In such ambiguous cases, we assume the worst case for the current configuration with respect to the incoming item. This simplifies the analysis in that we need not distinguish between different light bins according to their capacity (which would require distinguishing between many more different ranges of the incoming item sizes), yet it also complicates the analysis slightly in that we need to consider several possibilities (for the adversary's choice) at each step. These possibilities are unavoidable in that we can't *a priori* decide which one would be the worst case.

Now let's describe the way to compute the worst-case value of $E\big[\sum_{t=1}^{C}\big(s_{m+2}(t)+s_{m+3}(t)+\cdots+s_{j+1}(t)\big)/C \,\big|\, \text{config. at time } 0\big]= E\big[\sum_{t=1}^{C} X_t/C \,\big|\, \text{config. at time } 0\big]$ over C steps, given that an oblivious adversary is making choices at each step of the process. Let

$$E_Q(S) = E\Big[\sum_{t=1}^{Q} X_t \,\Big|\, \text{config. at time } 0 \text{ is } S = (X_0, A_0, B_0)\Big],$$

be the worst-case expected value of $\sum_{t=1}^{Q} X_t$, if we run the process for Q steps starting from configuration $S = (X_0, A_0, B_0)$.

1. Start by computing the $E_1(S)$ for all $S = (X_0, A_0, B_0)$, with $0 \leqq X_0 < C$. This is easy to do, because for each adversarial choice we can see which yields a smaller expected value of X_1.
2. Proceed inductively for $0 < i \leqq C$. Compute $E_i(S)$ for all S, given knowledge of $E_{i-1}(S)$ for all S. We always pick the adversarial choice which yields a configuration S' with minimal $E_{i-1}(S')$.
3. Finally, we check whether $E_C(0,0,0)/C \geqq 1+\varepsilon'$, for some big enough $\varepsilon' > 0$ to absorb the floating-point error of the computation.

We run the above program for increasing values of C until we find the one that yields $E_C(0,0,0)/C \geqq 1+\varepsilon'$.

Finally, note that we need to apply this analysis over the space of all (m, α) pairs. Suppose one focuses on a specific value of α (such as $\alpha = 33/50$), and proves $E_C(0,0,0) > 1+\varepsilon'$, by splitting up the possible values of m over a small constant number of ranges. The claim is then, that there are small constants $\delta, \varepsilon'' > 0$ such that $E_C(0,0,0) > 1 + \varepsilon''$ for $\alpha' \in [\alpha - \delta, \alpha + \delta]$. Hence, it suffices to try a sufficiently dense subset of α values in the range $[33/50, 2/3)$, and by the "continuity" from the claim, conclude that $E_C(0,0,0) > 1 + \varepsilon$ for a suitable ε everywhere in the interval.

Proof of claim: Let's look at two settings, one with $j = \alpha k$ and one with $j = \alpha' k$, such that $\alpha \leqq \alpha' \leqq \alpha + \delta$. Let $p_t(x)$ and $p'_t(x)$ be the probability of $X_t = x$ in the worst case in both settings, respectively. Also, denote by $p_q(w \leqq \alpha)$ the probability that the incoming items for the first q steps in the second setting are always with sizes not bigger than αk. We have

$$p'_t(x) = p_t(w \leqq \alpha) p_t(x) + \big(1 - p_t(w \leqq \alpha)\big) h_t(x), \tag{23}$$

for some $h_t(x) \leqq 1$. We know that $p_t(w \leqq \alpha) = \big(\alpha/\alpha'\big)^t$. Next we look at

$$\begin{aligned} p_t(x) - p'_t(x) &= p_t(x) - p_t(w \leqq \alpha)p_t(x) - \big(1 - p_t(w \leqq \alpha)\big)h_t(x) \\ &= \big(p_t(x) - h_t(x)\big)\big(1 - p_t(w \leqq \alpha)\big) \\ &\leqq 1 - p_t(w \leqq \alpha) \\ &= 1 - \left(\frac{\alpha}{\alpha'}\right)^t \\ &\leqq 1 - \left(\frac{\alpha}{\alpha + \delta}\right)^t \\ &= \bar{\varepsilon}_t(\alpha, \delta). \end{aligned} \tag{24}$$

We look at the expectations of X_t and X'_t for $1 \leqq t \leqq C$

$$\begin{aligned} E[X_t] - E[X'_t] &= \sum_{i=0}^{C-1} i\big(p_t(x) - p'_t(x)\big) \\ &\leqq C \sum_{i=0}^{C-1} \big(p_t(x) - p'_t(x)\big) \\ &\leqq C \sum_{i=0}^{C-1} \bar{\varepsilon}_t(\alpha, \delta) \\ &\leqq C \sum_{i=0}^{C-1} \bar{\varepsilon}_C(\alpha, \delta) \\ &\leqq C^2 \bar{\varepsilon}_C(\alpha, \delta). \end{aligned} \tag{25}$$

Hence,

$$\begin{aligned} &E\left[\sum_{t=1}^{C} X_t/C \,\middle|\, \text{config. at time } 0\right] - \\ &\quad E\left[\sum_{t=1}^{C} X'_t/C \,\middle|\, \text{config. at time } 0\right] \leqq C^2 \bar{\varepsilon}_C(\alpha, \delta) \\ &\qquad = C^2 \left[1 - \left(\frac{\alpha}{\alpha + \delta}\right)^C\right]. \end{aligned} \tag{26}$$

That means that the expectation of the sum of the X_t's is continuous in α. $\square$

The list of choices that the adversary can take at each step according to the range of the incoming item size is big enough not to be included here. (A sample case is included in the appendix.) Calculating up to E_{100} is sufficient to show that supersteps of 100 steps end up with an increase of s_1 in most of the cases. (The worst case appears to be the lower end of the interval, 33/50; and the result cannot be extended beyond 2/3 simply because the open range lemma ceases to apply.) The lower barrier 33/50 is chosen for convenience. It appears that additional work on detailing cases would be necessary to extend the result below $0.65 = 13/20$.

3.7 Tying together steps and supersteps

So far we've shown that we can think of the Best Fit bin packing process as a sequence of steps and supersteps, such that the steps tend to not decrease s_1, whereas the (constant length) supersteps tend to increase it slightly, i.e. $E\left[\sum_{t=1}^{C} X_t/C \,\middle|\, \text{config. at time } 0\right] > 1+\varepsilon$. This in itself makes the process equivalent to a random walk with a probability of increasing not less than the probability of decreasing. In other words, it already shows that the process is diverging. The next step is to show that it is diverging in a strictly linear fashion.

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

Proof: Let Y_t be the indicator function of the event $s_1(t) \neq 0$, and define $Z_t = X_t - Y_t$. The distribution of $s_1(t)$ given the state at time $t-1$ is:

$$s_1(t) = \begin{cases} s_1(t-1)+1, & \text{w.p. } X_{t-1}/j, \\ s_1(t-1)-1, & \text{w.p. } Y_{t-1}/j, \\ s_1(t-1), & \text{otherwise.} \end{cases} \tag{27}$$

Thus we get:

$$\begin{aligned} E\left[s_1(t) \| s_1(t-1), Z_{t-1}\right] &= \frac{X_{t-1}}{j}\big(s_1(t-1)+1\big)+ \\ &\quad + \frac{Y_{t-1}}{j}\big(s_1(t-1)-1\big)+\left(1-\frac{X_{t-1}}{j}-\frac{Y_{t-1}}{j}\right)s_1(t-1) \\ &= s_1(t-1)+\frac{Z_{t-1}}{j}. \end{aligned} \tag{28}$$

Therefore we have $E\left[s_1(t)\|Z_{t-1}\right] = E\left[s_1(t-1)\right] + Z_{t-1}/j$, and consequnetly $E\left[s_1(t)\right] = E\left[s_1(t-1)\right] + (1/j)E\left[Z_{t-1}\right]$; and from the latter, by induction, we obtain:

$$E\left[s_1(t)\right] = \frac{1}{j}\sum_{i=0}^{t-1} E\left[Z_i\right] + s_1(0). \tag{29}$$

We always have $Z_t \geqq -1$; in fact, the difficult configurations are precisely when $Z_t = -1$. Consider running the (general) Markov chain for n steps, starting from some given initial configuration. Divide time into *supersteps* in the following way. A superstep is simply a normal step of the chain, except in the case when we are in a state with $Z_t = -1$. In this case, all the steps from this point until the stopping time are combined into a superstep; call the latter *long* *supersteps* and the former, respectively, *short supersteps*. Every short superstep has $Z_t \geqq 0$ and every long superstep has $E\left[\sum_{t\leqq\tau} Z_t\right] > 0$, where τ is the stopping time of the long superstep.

To show linear waste, we are going to demonstrate that $E\left[s_1(t)\right]$ is linear in t. Specifically, we show that for some big fixed n_0, over any n_0 subsequent steps of the chain, the expected change of s_1, $E\left[\Delta s_1(n_0)\right]$, is constant (fraction of n_0); and then apply $E\left[s_1(t)\right] = t/n_0 E\left[\Delta s_1(n_0)\right]$. We want to check that over the n_0 steps the number of short supersteps with $Z_t > 0$ plus the number of long supersteps is not less than cn for some fixed $c \in (0, 1]$ [4].

We have already shown that long supersteps tend to increase s_1 and short supersteps do so too, if $Z_t > 0$; all we need to show now is that it is very unlikely to have a situation in which almost all of the supersteps within a sequence of n_0 chain transitions are short with $Z_t = 0$. This may appear obvious; however, it is *a priori* possible that all of the s_i except those determined in the open range lemma are greater than 0 for almost all steps, in which case we might have $Z_t = 0$ for almost every (short) superstep.

We show that this is not the case, simply by showing that the area adjacent to the open range guaranteed by the open range lemma must also be open a constant fraction (of n_0) of the time. Spceifically, set $\gamma = 2/3 - \alpha$. We show that all of $s_{(1-\alpha)k}, s_{(1-\alpha)k+1}, \ldots, s_{(1-\alpha+\alpha\gamma/2)k}$ are simultaneously 0 for an expected constant fraction of the time steps. From this it is easy to conclude that an expected constant fraction of the steps are short supersteps with $Z_t > 0$, since from a state where $s_{(1-\alpha)k}, s_{(1-\alpha)k+1}, \ldots, s_{(1-\alpha+\alpha\gamma/2)k}$ are all simultaneously 0 we achieve a state where exactly two non-neighbouring s_i in the range $\left[(1-\alpha)k, (1-\alpha+\alpha\gamma/2)\,k\right]$ are non-zero in two time steps with constant probability. In fact the probability is approximately $\gamma^2/4$, corresponding to the immediate insertion of two items into empty bins that yield bins with remaining capacity in this range (note that this probability estimate is not violated if there is a light bin, and as before problematic events happen with negligible probability so they don't affect the analysis). To sum up, an expected constant fraction of configurations have the property that there are no bins within the range of capacities $\left[(1-\alpha)k, (1-\alpha+\alpha\gamma/2)\,k\right]$, and some constant fraction of them produce short supersteps with $Z_t = 1$ or 2 in two steps. Of course, some of these configurations may fall within long supersteps in which case we don't consider them, because the long superstep already ensures that we are tending to increase s_1. So if we prove that bn_0 of the configurations have the adjacent range property, then we would have at least $(b/C)n_0$ supersteps (long or short) which tend to strictly increase s_1.

To show that all of $s_{(1-\alpha)k}, s_{(1-\alpha)k+1}, \ldots, s_{(1-\alpha+\alpha\gamma/2)k}$ are simultaneously 0 for an expected constant fraction of the time steps, we show that $S = s_{(1-\alpha)k} + s_{(1-\alpha)k+1} + \cdots + s_{(1-\alpha+\alpha\gamma/2)k}$ is stochastically dominated by a random walk that is biased downward. This follows since there are at most $(\alpha\gamma/2)k$ entering item sizes that increase S when $S \neq 0$ [5], corresponding to when items are placed into empty bins. No other items sizes can increase S because all items sizes which could possibly make a bin of capacity $> (1-\alpha+\alpha\gamma/2)k$ into a bin of capacity $\in \left[(1-\alpha)k, (1-\alpha+\alpha\gamma/2)\,k\right]$ would have to fall in of the bins in S, since we assumed $S \neq 0$. On the other hand, when the range $s_{k/3}, \ldots, s_{k-j-1}$ are all 0, at least $\gamma k - 2$ (which is bigger than $(\alpha\gamma/2)k$ as k grows) possible item sizes decrease S; namely, any item size in the range $\left[k/3, k-j-1\right]$.

Hence, we have shown that each chain of n_0 steps will contain at least cn_0 supersteps, each of which is either short with $Z_t > 0$ or long. Therefore $E\left[s_1(t)\right] = \Theta(T)$. Also, the constant factor

[4] It can happen sometimes that the n_0 boundary cuts some long superstep in the middle, but this is not going to be crucial for our analysis, since we can choose n_0 big enough so that $cn_0 > C$.

[5] We are only interested in the case when $S \neq 0$, because when $S = 0$, S can only stay constant or go up.

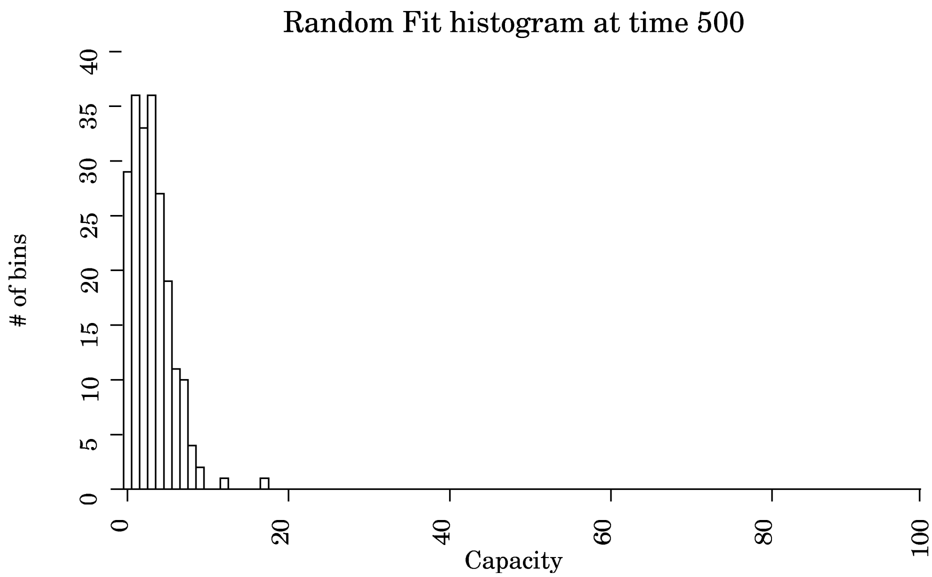


Figure 3: This histogram is similar to the one for Best Fit on Figure 1. Notice, however, that with Random Fit s_1 is not so great, and is even less then s_2, s_3 and s_4.

implied by the $\Theta(\cdot)$ notation is in fact independent of k.

4 Analysis of discrete skewed distributions of Random Fit

This section presents a discussion of the conjectured divergence of Random Fit, using ideas from the proof of the Best Fit case.

4.1 The general Markov chain

The general Markov chain in the case of Random Fit has the same states as in Best Fit, but different transition probabilities. Let l be the size of the next item inserted.

Let $i_1 < i_2 < \cdots < i_p$ be the maximal sequence of indices of bins whose capacity is equal or bigger than l, if such exists; in that case, Random Fit inserts item l into a bin with capacity i_σ, where $\sigma \stackrel{R}{\leftarrow} \{1, \ldots, p\}$, so, if the capacity of the bin with index i_σ was c, after insertion we get $s_c(t+1) = s_c(t) - 1$ and, if $c > l$, $s_{c-l}(t+1) = s_{c-l}(t) + 1$; all other components of $s(t)$ remain unchanged. If no such sequence exists, then the algorithm inserts item l into an empty bin, so we have $s_{k-l}(t+1) = s_{k-l}(t) + 1$ and all other components of $s(t)$ remain unchanged. This completes the description of the Markov chain.

4.2 The difficult configurations

The case with Random Fit is a bit different then Best Fit. With Best Fit we had that the probability of increasing s_1 is X_t/j and the probability of decreasing it is Y_t/j. With Random Fit, clearly, s_1 can decrease only when it is not 0; in this case, let $c_1 < c_2 < \cdots < c_q$ be the maximal sequence of indices, for which $s_{c_1} > 0, \ldots, s_{c_q} > 0$ and $c_1 = 1$. Then there is one way in which s_1 could potentially decrease, namely if an item of size 1 arrives and it gets packed in a bin of remaining capacity 1; in this case, the probability of decreasing would be $V_t = s_1/\big(j\sum_{i=1}^{q} s_{c_i}\big)$. On the other hand, (assuming that $j < k-1$) an incoming item of size $c_z - 1 \leqq j$ would increase s_1, if it gets packed in a bin of capacity c_z, which happens with probability $s_{c_z}/\big(\sum_{i=c_z-1}^{q} s_i\big)$, for $z \geqq 2$.

It follows that the total probability of increasing s_1 is

$$U_t = \frac{1}{j}\sum_{z=2}^{X'_t+1} \frac{s_{c_z}}{\sum_{i=c_z-1}^{q} s_i} \tag{30}$$

$$\geqq \frac{1}{j}\frac{1}{\sum_{i=1}^{q} s_{c_i}}\sum_{z=2}^{X'_t+1} s_{c_z} \tag{31}$$

where X'_t, the number of ways which can potentially increase s_1, is such that $c_1 < \cdots < c_{X'_t+1} \leqq j+1 < c_{X'_t+2} < \cdots < c_q$.

Classifying the difficult configurations in the case of Random Fit is slightly more subtle than in the Best Fit case. In the Best Fit case, a configuration was difficult if it had bins of capacites in all points of the range $[0, \xi k]$ for some $\xi < \alpha$, and no bins of capacities in the range $[\xi k, \alpha k]$. This classification has two strengths. First, it gives a definitive way of judging whether a configuration is difficult or not, in other words it is an "if and only if" classification. And second, it is simple to apply – it only refers to the presence of bins with capacities in certain fixed ranges, but does not refer to the number of bins within any range.

There is no such simple classification in the case of Random Fit, because one can virtually create both easy and difficult configurations for almost any "presence constraint." For instance, if I am given a configuration $\mathscr{C}$ and all I am told about it is that it has bins with capacities in certain given ranges, in most of the cases I won't be able to tell whether this is an easy or difficult configuration from this information. For example, consider the following two configurations which have bins of capacities 1 and 2 and no other bins, yet one of them is easy and one is difficult: $s_1 = 1, s_2 = 100$, or $s_1 = 100, s_2 = 1$ for any k. More surprisingly, we can create two configurations which have bins with capacities 1, γk, for $\gamma < \alpha$, and νk, for $\nu > \alpha$, and still one will be easy and one will be hard: $s_1 = s_{\gamma k} = s_{\nu k} = 1$; or $s_1 = 3$ and $s_{\gamma k} = s_{\nu k} = 1$.

The best we can do in the case of Random Fit is the following lemma:

Lemma 2. *Any Markov state of the general chain for which there exists $\nu \leqq \alpha$, such that $s_{\nu k} = 0$ and $s_{\nu k+1} \neq 0$, and $s_{\nu k+2} = \cdots = s_{k-1} = 0$, is easy in the sense that the probability of s_1 increasing on the next step is bigger than the probability of it decreasing.*

Proof. Let $c_1 < c_2 < \cdots < c_{X'_t+1}$ (where X'_t is as above) is the maximal sequence of capacities such that $s_{c_i} \neq 0$, for all $0 < i \leqq X'_t + 1$. Then the probability of decreasing s_1 is $V_t = s_1/\big(j\sum_{i=1}^{X'_t+1} s_{c_i}\big) < 1/j$. It is strictly less than $1/j$, because the condition from the lemma ensures that $s_1 < \sum_{i=1}^{X'_t+1} s_{c_i}$. On the other hand, let $c_w = \nu k + 1$, then probability of increasing s_1 would be:

$$U_t \geqq \frac{1}{j}\sum_{z=w}^{X'_t+1} \frac{s_{c_z}}{\sum_{i=c_z-1}^{q} s_i} \tag{32}$$

$$\geqq \frac{1}{j}\frac{1}{\sum_{i=w}^{q} s_{c_i}}\sum_{z=w}^{X'_t+1} s_{c_z} \tag{33}$$

$$= 1/j. \tag{34}$$

□

In view of this lemma and the examples given, we can naturally classify the general Markov chain states in four categories:

1. $s_1 \neq 0, \dots, s_m \neq 0$ and $s_{m+1} = \dots = s_{k-1} = 0$ for some $0 \leqq m < 1.3k$ (due to the Open Range Lemma). States in the category can be both difficult or easy as demonstrated in the examples above, and they could potentially be attacked by the methodology used in Best Fit's case.
2. $s_1 \neq 0, \dots, s_m \neq 0$, $s_{m+1} = \dots = s_{b-1} = s_{b+1} = \dots = s_{k-1} = 0$ and $s_b = 1$, for $b > \alpha k + 1$ and $0 \leqq m < 1.3k$. Just like in the previous category, states in the category can be both difficult or easy as demonstrated in the examples above, and they could potentially be attacked by the methodology used in Best Fit's case.
3. There exists $\nu \leqq \alpha$, such that $s_{\nu k} = 0$ and $s_{\nu k+1} \neq 0$, and $s_{\alpha k+2} = \dots = s_{k-1} = 0$. This category represents the definitely easy states (according to the above lemma).
4. There exists $\nu \leqq \alpha$, such that $s_{\nu k} = 0$ and $s_{\nu k+1} \neq 0$, $s_{\alpha k+2} = \dots = s_{b-1} = s_{b+1} = \dots = s_{k-1} = 0$ and $s_b = 1$ for $b > \alpha k + 1$. This category represents states that are potentially difficult, and cannot be analyzed using techniques similar to those of the Best Fit case.

Examples of these categories are shown on Figure 4. Categories 1 and 2 are susceptible to analysis similar to that of the Best Fit case, because they are easily describable by a configuration scheme like the one used in for Best Fit. For example, if we were to use the Best Fit notation, Category 1 states would correspond to $(0, 0, 0)$ and Category 2 states would correspond to $(0, 0, 1)$. In the next sections I explain why it is still difficult to use that type of analysis to prove divergence for these two categories. Category 3 represents easy states, according to the lemma, so we need not take care of it. In a later section I am going to show that most states in Category 4 are easy, and that the few ones that are not are susceptible to case analysis.

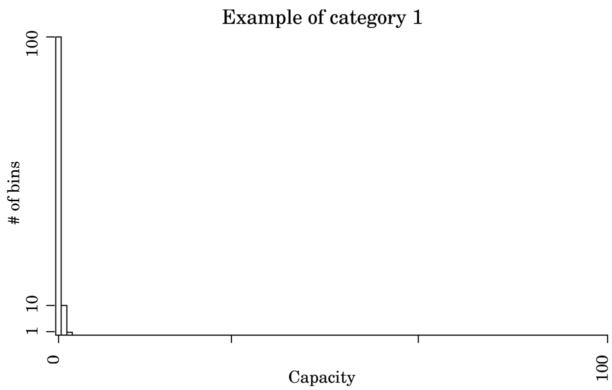
4.3 A Markov chain of supersteps

This and the next section will be devoted to dealing with difficult states of Category 1 or 2. Again, we shall take advantage of the superstep setting. As soon as the process goes into a difficult state of Category 1 or 2, we are marking the beginning of a superstep. The stopping time framework would be the same as in the Best Fit case, which will ensure that all of the following holds in most of the time:

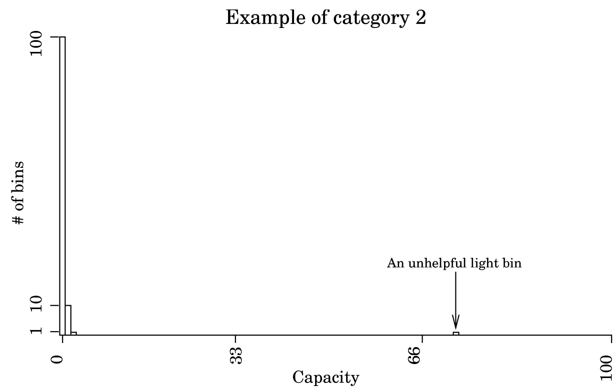
1. At any moment in time during the superstep, the general Markov state of the system is of the form $s_1 \neq 0, \dots, s_m \neq 0$ and $s_{u_1} \neq 0, \dots, s_{u_q} \neq 0$ and all other s_i's are zero, where there are no two consequtive numbers in the sequence $u_1, \dots, u_q$,and $u_1 > m$.
2. m stays constant throughout the superstep.
3. The superstep takes exactly C steps.

In general, any set of problematic events could potentially be used as long as we ensure that as k grows to infinity the probability of a problematic event becomes negligible. And, in fact, we only need to make sure that for all $k > k_0$, for some $k_0 \in \mathbb{N}$, the probability of a problematic event is less then some constant p_o, which is small enough to ensure that s_1 tends to increase even when problematic events happen, and to dampen the effect of the floating point imprecision of the computer we are using for our analysis.

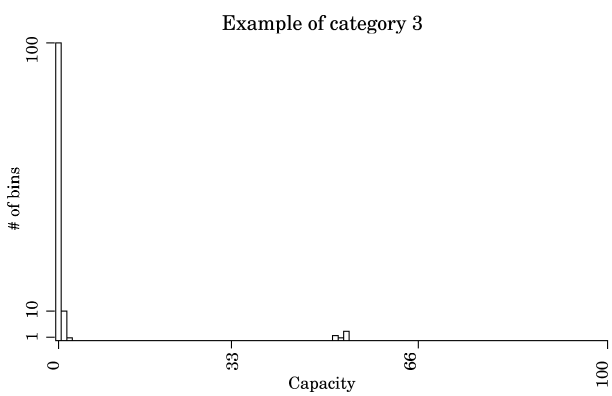
More specifically, we are hoping that at the end of the analysis we shall be able to prove a statement like this: "the difference between the probability of s_1 having increased and s_1



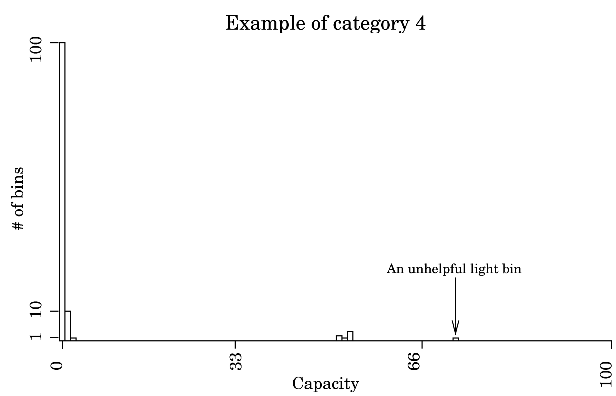
Category 1: $s_1 \neq 0, \ldots, s_m \neq 0$ and $s_{m+1} = \cdots = s_{k-1} = 0$.



Category 2: $s_1 \neq 0, \ldots, s_m \neq 0$, and $s_{m+1} = \cdots = s_{b-1} = s_{b+1} = \cdots = s_{k-1} = 0$ and $s_b = 1$ for $b > \alpha k + 1$.



Category 3: There exists $\nu \leqq \alpha$, such that $s_{\nu k} = 0$ and $s_{\nu k+1} \neq 0$, and $s_{\alpha k+2} = \cdots = s_{k-1} = 0$.



Category 4: There exists $\nu \leqq \alpha$, such that $s_{\nu k} = 0$ and $s_{\nu k+1} \neq 0$, $s_{\alpha k+2} = \cdots = s_{b-1} = s_{b+1} = \cdots = s_{k-1} = 0$ and $s_b = 1$ for $b > \alpha k + 1$.

Figure 4: Categories 1 to 4 classify all possible general Markov chain states. Category 3 represents an easy state; categories 1 and 2 represent (generally) difficult states that can possibly be attacked using the case analysis for Best Fit; category 4 represents a (generally) difficult state which, I show, is equivalent to Categories 1 and 2. In all of the above $0 \leqq m < 1/3k$.

having decreased after the superstep terminates is greater than $d_* > 0$, on the condition that problematic events don't happen." Given such a statement, we would like to be able to prove that the probability of s_1 having increased after the termination of a superstep is more than the probability of it having decreased in the general setting, i.e. even when problematic events do happen. We also have to account for computational imprecision ϵ_c, so in the worst-case the difference will really be $d'_* = d_* - \epsilon_c$. We know that in the case of a problematic event the worst that can happen is s_1 to decrease, so we simply want to choose p_o such that $d'_* > p_o$.

We need to ask ourselves how much convenience could we possibly get by picking a good set of problematic events. From our discussion above, it follows that the set of problematic events could be any set of events who size is a very small constant fraction of k, but since we don't know d_* a priori and since p_o is so small, it is hard to come up with a useful set of events which has a size a fraction of k. Constant-sized sets of problematic events seem to be most useful. The set of events that we are using above makes sure that bins not in the range $[0, m]$ are never next to each other; one can easily extend this to a set of problematic events that ensures that these bins are never any closer than l from each other, for some constant l. This is so becase the set of events that violate this condition are linear in the number of steps C, which is constant. And so, even more generally, one can pick problematic events that ensure that some constant-sized vicinity of each bin not in the range $[0, m]$ never looks like any of a fixed number of "vicinity appearances." We could also ensure that certain fixed size (or small fraction sized) regions of the histogram never fall into any of a fixed number of states, and so on and so on. Unfortunately, none of these is really too useful for any analysis; all that problematic events really achieve for us is convenience when it comes to expressing the probabilities of s_1 increasing or decreasing, because these events take care of getting rid of unpleasant extreme cases that never happen.

Next, let's turn to expressing the probabilities of increasing and decreasing s_1 in a superstep. As before, let $A(t)$ be the probability that t is not the stopping time τ (on the condition that we have already reached step t and X_t is known, where X_t is taken to mean $s(m+1)+\cdots+s(\alpha k+1)$). Then the union bound tells us that $A(t) \geqq 1 - \sum_{i=1}^{7} T_i(t)$, where $T_i(t)$ is the probability that a stopping event of type i happens at time t; and since $T_i^{BF}(t) \leqq T_i(t) \leqq T_i^{BF}(t)/C$, where $T_i^{BF}(t)$ is the equivalent of $T_i(t)$ for Best Fit, we get:

$$A(t) = 1 - O(1/j^2) \tag{35}$$

Hence, the probability $\overline{P}$ that s_1 has increased would be

$$\begin{aligned} \overline{P} &= \sum_{t=1}^{C} \left[\frac{H_{X'_t}}{j} \prod_{i=1}^{t-1} A(i) \right] && (36) \\ &\geqq \sum_{t=1}^{C} \left[\frac{H_{X_t}}{j} \prod_{i=1}^{t-1} A(i) \right] && (37) \\ &= \frac{1}{j} \sum_{t=1}^{C} H_{X_t} + O\left(\frac{1}{j^2}\right), && (38) \end{aligned}$$

where X'_t is as in the previous two sections and $H_n = \sum_{i=1}^{n} 1/i$ is the harmonic number of n.

Again, using a union bound, we find the upper bound on the probability that s_1 has decreased (after the end of the superstep):

$$\underline{P} \leqq \frac{C}{j}. \tag{39}$$

This formula also assumes the worst-case that s_1 is so much bigger than $s_2 + \cdots + s_{k-1}$ that any incoming item of size 1 will virtually go into a bucket of capacity 1.

For the difference of the two probabilities, we get:

$$\overline{P} - \underline{P} \geqq \frac{\sum_{i=1}^{C} H_{X_i} - C}{j} + O\left(\frac{1}{j^2}\right). \tag{40}$$

$\overline{P}$, however, in the form shown above is, of course, a conditional probability which assumes the knowledge of all of $X_1, \ldots, X_C$. Exactly along the lines of the Best Fit analysis, we get the unconditional version:

$$\overline{P} = \frac{1}{j} E\left[\sum_{i=1}^{C} H_{X_i}\right] + O\left(\frac{1}{j^2}\right). \tag{41}$$

In other words, we need to show that

$$\overline{P} - \underline{P} = \frac{C}{j}\left[\frac{E\left[\sum_{i=1}^{C} H_{X_i}\right]}{C} - 1\right] + O\left(\frac{1}{j^2}\right) \tag{42}$$

$$\geqq 0, \tag{43}$$

or, as before, simply:

$$E\left[\frac{\sum_{t=1}^{C} H_{s_{m+2}(t)+s_{m+3}(t)+\cdots+s_{j+1}(t)}}{C} \,\middle|\, \text{configuration at time } 0\right] > 1 + \varepsilon'. \tag{44}$$

for some $\varepsilon' > 0$.

4.4 The configuration space

Before jumping into the specific configuration framework that I have picked for Random Fit, I'd like to say a few words about configuration spaces in general. We already mentioned that the configuration space size should be a constant rather than a growing function of k, because in the latter case it won't be easy to analyse and certainly it won't be possible to apply Kenyon and Mitzenmacher's case analysis.

The subtlety of the analysis is really in picking the right configuration space, because once this is done, the case-analysis is uniquely determined, in a sense. The analysis has to take into consideration all possibilities for α,w and m in order to be complete. Since these are continous variables, in practice, all we can do is consider small ranges of these variables, and write a case analysis for these ranges. The analysis is uniquely determined in the sense that, once we pick how finely grained our ranges for α,w and m will be, the list of cases is already etched in stone. When considering each specific case any ambiguity as to what may be the next state is turned into a choice for the adversary.

The Best Fit case has a relatively simple configuration space and yet it limits the ambiguity enough so that the tendency of increasing s_1 is clearly exhibited. This is also seen in the histogram for Best Fit on Figure 1. Unlike it, Figure 3 already hints that Random Fit's s_1 is much less upward biased.

My experiments have shown that a configuration space of the form (X_t, A_t, B_t), as in Figure 2, does not suffice for Random Fit because it allows for too much ambiguity and hence a lot of power for the adversary. This configuration space allowed me to prove that supersteps starting with states of Categories 1 and 2 end up increasing s_1 with prevailing probability for the cases when $m \in [0, 0.05475]$ and $m \in [0.33, 1/3]$. I used case analysis similar to the one for Best Fit. Since this proved to be insufficient, my next attempt was to enrich the configuration space.

I attacked the most difficult case, i.e. when $m = \xi k$ for $\xi = 1/6$, because the other cases would have easily followed from this one. The reason for considering $m = k/6$ to be the most difficult case is intuitively as follows. There are two other cases: when m tends to be closer to 0, or when m tends to be closer to 1/3. In the former case, the probability of any tokens falling in the area $[0, m]$ is very small and so most tokens contribute to adding more possibilities (X_t) for potentially increasing s_1. In the latter case, since m is so big, most of the time the incoming items don't result in decreasing X_t because they often fall in $[0, m]$, which helps bias the random walk upwards. The fact that this is the case can easily be seen in the case analysis for Best Fit (given in the appendix) which is essentially the same for Random Fit. The configurations $m = k/6$ are the most difficult to handle because they are a fine micture between the above two extremities.

I chose to use configurations of the form (Z_t, Y_t, A_t, B_t), where Z_t is the number of bins in the range $[m + 1, (1 - \alpha)k]$, Y_t is the number of bins in the range $[(1 - \alpha)k, k/2]$, A_t is 1 if there is a light helpful bin and 0 otherwise, and B_t is 1 if there is a light unhelpful bin and 0 otherwise. This is illustrated on Figure 5.

There are a number of reasons to pick this particular configuration space and not any other one:

1. By splitting what used to be X_t into Z_t and Y_t, we gain more information about the state underlying a configuration, and ultimately we are able to follow more closely the tokens' path on the histogram.
2. Why wouldn't we pick an even finer partition of X_t? This is a matter of taste. The size of the configuration space, and hence the running time of the analysis algorithm, is linear in C when we use X_t; it is quadratic in C when we use Z_t and Y_t; and the finer the partition get, the higher the exponent becomes. It takes about 1 hour to do the analysis (written in C++) in the case when we are using Z_t and Y_t and $C = 100$, which means that any finer partition would throw us into very lengthy computations.

   Similarly, one can ask the question: why not partition A_t and/or B_t? This would also increase the running time nearly twice.
3. Lastly: why did I pick the regions of Z_t and Y_t to be equally sized? There's to explanation to this. First, the intuitive explanation: imagine, if we were to gradually shrink Z_t's range (and thus expand Y_t's range), then we would converge to the case when we simply have X_t, because when Z_t's range is very small it has almost no impact on our analysis; and of course this argument is symmetric.

   Let's look at a specific example. Consider the two cases given on Figure 6. In the first one $l(Z_t) = l(Y_t)$, where $l(\cdot)$ represents the interval length of the range, and in the second one, say, $l(Y_t) = 2l(Z_t)$. Now, e.g. assume we are in a configuration $(0, 1, 0, 0)$ and an item of size $\xi k + 1$ comes. In the first case, the adversary will only have one option, namely: $X_{t+1} = 1$ and $Y_{t+1} = 0$; whereas, in the second case there are two options: $X_{t+1} = 1$ and $Y_{t+1} = 0$, or just $Y_{t+1} = 0$.

The actual case analysis is given in the next section. I think, however, that this is the the time to say what the results were. After I ran the analysis, using the enriched configuration space,

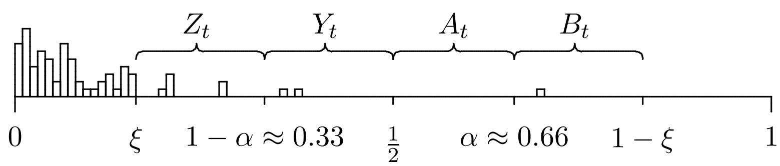


Figure 5: Regions of the Random Fit histogram, represented by the configuration (Z_t, Y_t, A_t, B_t). In that setting, there is no way to get any tokens past $(1-\xi)k$, hence B_t's range stops there.

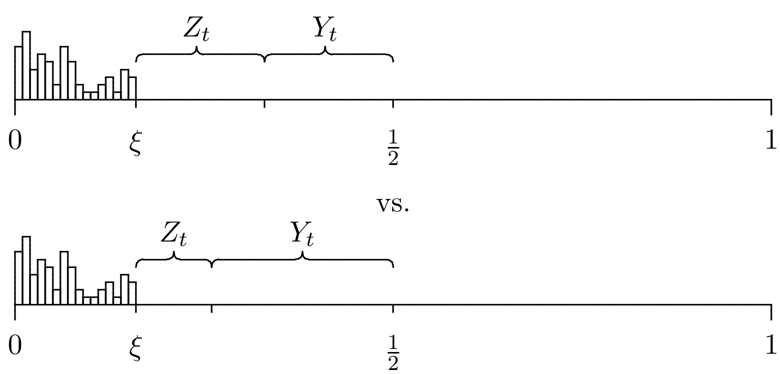


Figure 6: Different ways to partition X_t into Z_t and Y_t.

for 200 steps the expected value of $E\left[\sum_{t=1}^{C} H_{s_{m+2}(t)+s_{m+3}(t)+\cdots+s_{j+1}(t)}/C \,\middle|\, \text{conf. at time } 0\right]$ never went above 0.74, which is quite far from the goal of 1. Moreover, running the analysis for more steps wouldn't help wither because the expectation value converges to its equilibrium after the first 100 steps.

4.5 The dynamic program

Before giving the list of cases, I'd like to emphasize to observations that make this analysis different than the Best Fit analysis.

First, when an item of size $< \xi k$ comes, we give the adversary only one choice which is "no change." The reason for this is as follows: we can pick any number $\zeta < \xi$, and say that when an item of size $< \zeta k$ comes it gets packed in a bin of capacity $< \xi k$, because as $k \to \infty$ the size of the region $[\zeta k, \xi k]$ grows so large, that the probability that the incoming item gets packed there and nowhere else becomes effectively 1. And because we chose any $\zeta < \xi$, we might as well as that this holds for all incoming items of size $< \xi k$.

Second, suppose we are in a configuration $(10, 7, 0, 1)$ and an item of size $0.30k$ comes. In this case, we need to give the adversary two options: either pack the item in a bin with capacity in the range of Z_t or in a bin with capacity in the range of Y_t. The subtlety is that there choices have a limited probability, e.g. the first can happen in at most $1/(1 + Y_t + A_t + B_t)$ of the time, and the second choice can happen in at most $1/(Y_t + A_t + B_t)$ of the time (due to the nature of the Random Fit packing). In other words, the adversary has the power to decide what the positions of the tokens with the two ranges Z_t and Y_t are, but once that is fixed, it has no power to pack the incoming item in any bin it wishes to. This is reflected in my analysis, which follows:

* If $w \leqq \xi k$, no change.
* If $\xi k \leqq w \leqq 1 - \alpha$, the adversary has the choices:
  - $Z_{t+1} = Z_t - 1$, only if $Z_t > 0$. This choice can be exercised at most $Z_t/(Y_t + A_t + B_t + 1)$ of the time;
  - $Y_{t+1} = Y_t - 1$, only if $Y_t > 0$. This choice can be exercised at most $Y_t/(Y_t + A_t + B_t)$ of the time;
  - $A_{t+1} = 0$ and $Z_{t+1} = Z_t + 1$, only if $A_t = 1$. This choice can be exercised at most $1/(Y_t + A_t + B_t)$ of the time;
  - $A_{t+1} = 0$ and $Y_{t+1} = Y_t + 1$, only if $A_t = 1$. This choice can be exercised at most $1/(Y_t + A_t + B_t)$ of the time;
  - $B_{t+1} = 0$ and $A_{t+1} = 1$, only if $B_t = 1$. This choice can be exercised at most $1/(Y_t + A_t + B_t)$ of the time;
  - $B_{t+1} = 0$ and $Y_{t+1} = Y_t + 1$, only if $B_t = 1$. This choice can be exercised at most $1/(Y_t + A_t + B_t)$ of the time;
  - $B_{t+1} = 1$, only if $Y_t = A_t = B_t = 0$.
* If $1 - \alpha \leqq w \leqq 1/2$, the adversary has the choices:
  - $Y_{t+1} = Y_t - 1$, only if $Y_t > 0$. This choice can be exercised at most $Y_t/(A_t + B_t + 1)$ of the time;
  - $A_{t+1} = 0$ and $Z_{t+1} = Z_t + 1$, only if $A_t = 1$.
  - $A_{t+1} = 0$, only if $A_t = 1$.
  - $B_{t+1} = 0$ and $Z_{t+1} = Z_t + 1$, only if $B_t = 1$.

  - $B_{t+1} = 0$ and $Y_{t+1} = Y_t + 1$, only if $B_t = 1$.
  - $A_{t+1} = 1$, only if $A_t = B_t = 0$.
- * If $1/2 \leqq w \leqq \alpha$, the adversary has the choices:
  - $A_{t+1} = 0$, only if $A_t = 1$. This choice can be exercised at most $1/(A_t + B_t)$ of the time;
  - $B_{t+1} = 0$, only if $B_t = 1$.
  - $Y_{t+1} = Y_t + 1$, only if $B_t = 0$.

Figure 7 shows the pseudo-code of the analysis program.

4.6 Category 4

Let us say that we have just entered a state of Category 4: there exists $\nu \leqq \alpha$, such that $s_{\nu k} = 0$ and $s_{\nu k+1} \neq 0$, $s_{\alpha k+2} = \cdots = s_{b-1} = s_{b+1} = \cdots = s_{k-1} = 0$ and $s_b = 1$ for $b > \alpha k + 1$. W.l.o.g. assume that ν is the smallest number that possesses the above property, in which case $\nu k - 1 = m$. Assume nothing about the values of $s_1, \ldots, s_m$ and take the worst case when $\underline{P} = 1/j$, which is not even achievable in practice, if we are in Category 4. Denote $\tilde{S} = \sum_{i=\nu k+1}^{\alpha k+1} s_i$, then $\overline{P} \geqq \big(H_{\tilde{S}} + 1/\tilde{S} - 1/2\big)/j$. $\underline{P} > \overline{P}$ only when $\tilde{S} \leqq 5$, and hence each difficult state from Category 4 corresponds to some configuration which has $Y_t + Z_t = \tilde{S} \leqq 5$ and $A_t + B_t = 1$.

5 Conclusion

Could Random Fit's waste still diverge for the distributions $U\{j, k\}$, with $33k/50 < j < 2k/3$? The overarching philosophy of Kenyon and Mitzenmacher's approach is to take every difficult state and prove that a constant number of steps after it s_1 would have most likely increased. Even if we are to believe my guess that their analysis technique won't help prove Random Fit's divergence, there is still one possibility left. It might be the case that there exist difficult states which take non-constant number of steps to recover from, but these states might be very very unlikely to happen, which would alleviate the effect of their recovery time on s_1. Such a case would most likely require a different kind of proof technique.

References

[1] Problems in Analysis of Algorithms Pages. In *http://pauillac.inria.fr/algo/AofA/Problems/index.html*. Problem 5, posted by Ed Coffman.

[2] S. Albers and M. Mitzenmacher. Average case analysis of First Fit and Random Fit bin packing. In *Proceedings of the 9th Annual ACM Symposium on Discrete Algorithms*, pp. 290–299, 1998.

[3] C. Kenyon, Y. Rabani and A. Sinclair. Biased random walks, Lyapunov functions, and stochastic analysis of Best Fit bin packing. *Journal of Algorithms*, 27(2), pp. 218–235, 1998.

[4] Claire Kenyon and Michael Mitzenmacher. Linear waste of Best Fit bin packing on skewed distributions. In *Foundations of Computer Science*, 2000.

[5] D. Hochbaum, editor. *Approximation algorithms for NP-hard problems.* PWS Publishing Company, Boston, MA, 1997.

```
pre-initialized data
   For each of the case-analysis' ranges r̄ of w we have an array r̄[·] of
   triples of the form ⟨condition (D),action (A),maximal probability (M)⟩;

variables
   [A set of variables to hold the expected value of the superstep on
   the condition that we are in a given conf. at a given step]
   set of float vars named E indexed by t ∈ [0, C] and 𝒞 ∈ Ψ_t;

begin

   [initialize expectations given that we have 0 steps left]
   for 𝒞 ∈ Ψ_C begin
      E(C, 𝒞) := H_{X_𝒞 + A_𝒞};
   end

   [dynamic program]
   for t := C − 1 downto 0 begin
      for 𝒞 ∈ Ψ_t begin
         E(t, 𝒞) := H_{X_𝒞 + A_𝒞};
         for r̄ in set of ranges of w from case analysis begin
            Create a temporary copy t̄[·] of r̄[·];
            Eliminate all triples from t̄[·] whose condition is  not satisfied by 𝒞;
            Sort the remaining of t̄[·] by expectation of the
               next step configuration in ascending order;

            var float u = 0;
            for c := 1 to l(t̄[·]) begin
               E(t, 𝒞) := E(t, 𝒞) + E(t + 1, 𝒲) min{1 − u, t̄[c].M}l(r̄)/α where
                  𝒲 ∈ Ψ_{t+1} is 𝒲 = t̄[c].A(𝒞);
               u := u + min{1 − u, t̄[c].M};
            end
         end
      end
   end

end
```

Figure 7: Pseudo-code for the dynamic program of Random Fit's analysis. Here $l(\bar{r})$ represents the length of the range $\bar{r}$ in the sense of an interval length, and $l(\bar{t}[\cdot])$ is the number of elements in the array $\bar{t}[\cdot]$. At the end of the execution of this program, the variable $E(0,(0,0,0))$ holds the value of $E\left[\sum_{t=1}^{C} H_{X_t} \,\middle|\, \text{config. at time } 0\right]$, where $X_t = Z_t + Y_t + A_t$.

[6] 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 Journal on Computing*, 3, pp. 299–325, 1974.

[7] 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 of the 23rd Annual ACM Symposium on Theory of Computing*, pp. 230–240, 1991.

[8] 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. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pp. 412–421, 1993.

[9] E.G. Coffman Jr., M.R. Garey, and D.S. Johnson. *Approximation algorithms for bin packing: A survey.* In D. Hochbaum [5], 1997.

[10] Martin Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* John Wiley and Sons, Inc., 1994.

[11] Moshe Shaked and J. George Shanthikumar. *Stochastic Orders and their Applications.* Probability and Mathematical Statistics. Academic Press Limited, United Kingdom, London, 1994.

A Best Fit adversarial case analysis

In this appendix section, I'm giving a precise description of the dynamic program's operation with respect to the analysis of the Best Fit case. First of all, given that any difficult configuration is of the form $(0,0,0)$, i.e. this is the entering configuration of a superstep, and since we are only looking at C consequent steps, all configurations, reachable from within the superstep, are of the form (X,A,B), where $0 \leqq X \leqq C$, which gives a total of $3(C+1)$ different configurations.

In order to compute the quantity $E\big[\sum_{t=1}^{C} X_t/C \,\big|\, \text{config. at time } 0\big]$, we need to find the probability distributions on all of the following spaces of configurations Ψ_t, where $\Psi_t = \big\{(X,A,B) : 0 \leqq X \leqq t, (A,B) \in \{(0,0),(1,0),(0,1)\}\big\}$ represents all configurations reachable at time t from the beginning of the superstep. This is so because we are working on the condition that problematic events don't occur, and so the probability distribution on each of Ψ_t, for $0 \leqq t \leqq C$, should sum up to 1. Our computation of these probability distributions will be based on our knowledge of the distribution on Ψ_0, namely $\Pr_{\Psi_0}[(0,0,0)] = 1$ and $\Pr_{\Psi_0}[(0,1,0)] = \Pr_{\Psi_0}[(0,0,1)] = 0$.

If, given that we are in configuration $\mathscr{C}_t$ at time t, we could determine the probability distribution of our next step, we could easily compute all distributions on the Ψ_t's by starting from Ψ_0 and progressively going up to Ψ_C. Unfortunately, since a configuration does not contain complete information about the specific state of the system, a pair of a configuration and an incoming item does not determine uniquely the next step configuration. In other words, there is some ambiguity, which however is limited, because not all possible next step outcomes will be reachable.

Take for example the following scenario. Suppose we are already t steps into a superstep with $m = 0.1k$, and we know that we are currently in configuration $(1,0,0)$, and that the incoming item is in the range $[0.2k, 0.3k]$. There are two possibilities:

a. Either the incoming item is bigger than the capacity of the only useful bin ($X_t = 1$), in whichcase we are going into configuration $(1,0,1)$;

b. or not, in which case there are still two possible outcomes: $(0,0,0)$ or $(1,0,0)$, depending on what the difference between the incoming item's capacity and the useful bin's capacity.

In summary, given our current configuration and the range of the incoming item, we could end up in any one of three configurations and we have no way of telling which one it will be. Since there is ambiguity involved, the best we can do is compute the worst-case outcome of this situation. For this purpose, we need to be able to tell which choice will yield the worst case overall. And specifically in our example, we need to know which one of $(0,0,0)$, $(1,0,1)$ or $(1,0,0)$ yields the worst overall value of $E\left[\sum_{t=1}^{C} X_t/C \,\middle|\, \text{config. at time } 0\right]$ given that $C-t-1$ steps will be remaining. Hence, before making our choices for step t we should already know what the superstep outcome would be, if we were in any one of the configurations reachable in time $t+1$ and had $C-t-1$ steps remaining. To achieve this, we need to compute the distributions on Ψ_C first, then on Ψ_{C-1}, and so on to Ψ_0.

The following is the actual case analysis from [4]. The analysis is specific to the case where $1/4k \leqq m \leqq 1/3k$. Let the entering item have weight wk, where $0 \leqq w \leqq \alpha$. Recall that X_t is the number of ways to increase s_1; A_t is 1 if and only if there is helpful light bin, and 0 otherwise; B_t is 1 if and only if there is an unhelpful light bin, and 0 otherwise.

Case 1: $A_t = 0, B_t = 0$.

- If $w \leqq 1/4$, no change.
- If $1/4 \leqq w \leqq 1/3$, the adversary decides between setting X_{t+1} to $X_t - 1$ (if possible, i.e. if $X_t > 0$) or setting B_{t+1} to 1.
- If $1/3 \leqq w \leqq 1-\alpha$:
  * If $X_t = 0$, set B_{t+1} to 1.
  * If $X_t > 0$, the adversary decides between setting X_{t+1} to $X_t - 1$ or setting B_{t+1} to 1.
- If $1-\alpha \leqq w \leqq 1/2$:
  * If $X_t = 0$, set A_{t+1} to 1.
  * If $X_t > 0$, the adversary decides between setting X_{t+1} to $X_t - 1$ or setting A_{t+1} to 1.
- If $1/2 \leqq w \leqq \alpha$, set X_{t+1} to $X_t + 1$.

Case 2: $A_t = 1$.

- If $w \leqq 1/4$, no change.
- If $1/4 \leqq w \leqq 1/3$, the adversary decides between setting X_{t+1} to $X_t - 1$ (if possible) or setting A_{t+1} to 0 (either increasing X_t or not).
- If $1/3 \leqq w \leqq 1/2$:
  * If $X_t = 0$, set A_{t+1} to 0, and adversary decides whether to increase X_t or not.
  * If $X_t > 0$, the adversary decides between setting A_{t+1} to 0 (either increasing X_t, or not) and setting X_{t+1} to $X_t - 1$.

If $1/2 \leqq w \leqq \alpha$, the adversary decides between setting X_{t+1} to $X_t + 1$ or setting A_{t+1} to 0.

Case 3: $B_t = 1$.

- If $w \leqq 1/4$, no change.
- If $1/4 \leqq w \leqq 1/3$, the adversary decides between setting X_{t+1} to $X_t - 1$ (if possible) or setting B_{t+1} to 0 (either increasing X_t or not).

```
variables
  [A set of variables to hold the expected value of the superstep on
  the condition that we are in a given conf. at a given step]
  set of float vars named E indexed by t ∈ [0,C] and 𝒞 ∈ Ψ_t;

begin

  [initialize expectations given that we have 0 steps left]
  for 𝒞 ∈ Ψ_C begin
    E(C,𝒞) := X_𝒞 + A_𝒞;
  end

  [dynamic program]
  for t := C − 1 downto 0 begin
    for 𝒞 ∈ Ψ_t begin
      E(t,𝒞) := X_𝒞 + A_𝒞;
      for r̄ in set of ranges of w from case analysis begin
        E(t,𝒞) := E(t,𝒞) + E(t + 1,𝒲)l(r̄)/α where
          𝒲 ∈ Ψ_{t+1} is the one configuration, among the reachable
          ones from 𝒞 on an incoming item in the range r̄, that has
          the smallest corresponding E(t + 1,𝒲);
      end
    end
  end

end
```

Figure 8: Pseudo-code for the dynamic program of Best Fit's analysis. Here $l(\bar{r})$ represents the length of the range $\bar{r}$ in the sense of an interval length. At the end of the execution of this program, the variable $E(0,(0,0,0))$ holds the value of $E\left[\sum_{t=1}^{C} X_t \,\middle|\, \text{config. at time } 0\right]$.

– If $1/3 \leqq w \leqq 1/2$:
  * If $X_t = 0$, set B_{t+1} to 0, and adversary decides whether to increase X_t or not.
  * If $X_t > 0$, the adversary decides between setting B_{t+1} to 0 (either increasing X_t, or not) and setting X_{t+1} to $X_t - 1$.
– If $1/2 \leqq w \leqq \alpha$, set B_{t+1} to 0.

Similar analysis applies for the ranges of m: $0 \leqq m \leqq 12/100$, $12/100 \leqq m \leqq 22/100$, and $22/100 \leqq m \leqq 25/100$. Figure 8 provides the pseudo-code of the program that computes the worst-case $E\left[\sum_{t=1}^{C} X_t/C \,\middle|\, \text{config. at time } 0\right]$ for any case analysis of the above form.