AN AVERAGE-CASE ANALYSIS OF BIN PACKING WITH UNIFORMLY DISTRIBUTED ITEM SIZES

George S. Lueker+

Technical Report #181

February 1982

Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92717

The results in this report were presented under the title "Heuristics for Bin-Packing" at the Special Session on Theoretical Computer Science of the 791st Meeting of the American Mathematical Society, Santa Barbara, November 1981.

+ Supported by the National Science Foundation under grant MCS79-04997.

AN AVERAGE-CASE ANALYSIS OF BIN PACKING WITH UNIFORMLY DISTRIBUTED ITEM SIZES

George S. Lueker

Abstract

We analyze the one-dimensional bin-packing problem under the assumption that bins have unit capacity, and that items to be packed are drawn from a uniform distribution on [0,1]. Building on some recent work by Frederickson, we give an algorithm which uses $n/2+O(n^{1/2})$ bins on the average to pack n items. (Knödel has achieved a similar result.) The analysis involves the use of a certain 1-dimensional random walk. We then show that even an optimum packing under this distribution uses $n/2+\Omega(n^{1/2})$ bins on the average, so our algorithm is asymptotically optimal, up to constant factors on the amount of wasted space. Finally, following Frederickson, we show that two well-known greedy bin-packing algorithms use no more bins than our algorithm; thus their behavior is also in asymptotically optimal in this sense.

1. Introduction

We consider the following problem. Given n numbers $x_1,x_2,\ldots,x_n$, which represent weights, pack them into a minimum number of bins so that no bin has a total weight exceeding 1.

The worst-case behavior of algorithms for this problem has been the subject of considerable investigation. For a long time, the best-known worst-case asymptotic error bound was 11/9 [Jo73, JDUGG74]. This was improved very slightly by Yao [Ya80], and then improved to 71/60 by Johnson [GJ80]. Recently an algorithm with an asymptotic bound of $1+\varepsilon$, for any $\varepsilon>0$, has been obtained [VL81]. Unfortunately, all of these results have drawbacks from a practical standpoint. The algorithm in [VL81], while theoretically linear, has a huge constant for small values of ε. The earlier algorithms have error bounds which might be larger than we would desire.

A number of the early papers [Jo74, JDUGG74] suggested that an average-case analysis of the problem would be interesting. As observed in [CSHY80], analyses of algorithms for this problem can quickly become very complicated; there a next-fit strategy is analyzed under a rather general distribution of the item sizes x_i. For the case in which the sizes are uniformly drawn from [0,1], this strategy tends to leave the bins about 1/4 empty. Frederickson has shown that a different algorithm tends to waste much less space. Assuming that the x_i are drawn from a uniform distribution on [0,1], he gives an algorithm which uses an average of $n/2+O(n^{2/3})$ bins, and thus tends asymptotically to fill the bins almost completely. One easily sees that an average of at least n/2 bins will be required, since this is the expected total of the x_i. Thus Frederickson has established that the expected number of bins required is asymptotic to n/2. It is interesting, however, to look at the number of <u>extra</u> bins required beyond the sum of the x_i; as observed in [Sh77], this is equivalent to looking at the expected amount of wasted space in the packing. Frederickson's algorithm has an expected wasted space of $O(n^{2/3})$. Here we present an algorithm with an expected wasted space of $\Theta(n^{1/2})$; a similar result was achieved by Knödel [Kn81]. Moreover, we show that even an optimum packing wastes $\Theta(n^{1/2})$ space on the average. Thus in some sense our algorithm can not be improved, except for constant factors.

Our analysis of this algorithm will use some facts about sums of random variables, which are well-known or easily established.

Fact 1. Let $V_1, V_2, \ldots, V_k$ each have an exponential distribution with mean one; i.e., each has the density function e^{-x}, for $x>0$. Then, for any $\rho<1$, the probability that the sum of the V_i is less than or equal to ρk is exponentially small in k. (This is a special case of Theorem 1 in [Ch52]; such theorems are referred to as theorems about large deviations.)

The next fact is more interesting. The random variables considered will have distributions which are symmetric about the origin; instead of bounding only the sum of all the random variables, we wish to bound all of the partial sums. Thus we wish to bound the probability that a k-step random walk about the origin ever passes some point x.

Fact 2. Let $W_1, W_2, \ldots, W_k$ each have a bilateral exponential distribution; i.e., the density function for each W_i is $\frac{1}{2}e^{-|x|}$. Let $F^{*k}(x)$ be the cumulative probability distribution for the sum of k such variables; i.e.,

$$F^{*k}(x) = P\{W_1 + W_2 + \ldots + W_k \leq x\}.$$

Then the probability that any of the partial sums $W_1+W_2+\ldots+W_i$, $1\leq i\leq k$, exceed x is less than or equal to $2(1 - F^{*k}(x))$. (This is a special case of the Lévy inequalities [CT78, Section 3.3, Lemma 5, page 71].)

The next two facts have to do with expected values of quantities related to sums of random variables.

Fact 3. Let f(x) be any density function which satisfies the conditions for [Fe66, Chapter XVI.2, Theorem 2, page 508], with r=4. Let f have mean zero and variance σ^2, and let f^{*n} denote the pdf for the sum of n independent draws with pdf f. Then for any fixed $\alpha>0$,

$$\int_0^{\alpha n} x\, f^{*n}(x)\, dx = \sigma\, (\frac{n}{2\pi})^{1/2} + o(n^{1/2}).$$

Fact 4. $$\sum_{i=\lceil n/2 \rceil}^{n} 2^{-n} \binom{n}{i} (i-n/2) = (\frac{n}{8\pi})^{1/2} + o(n^{1/2}).$$

Verifications of Facts 3 and 4 are sketched in the Appendix.

2. The algorithm and its analysis

We begin by reviewing Frederickson's algorithm [Fr80], which forms the basis for our work; α is a parameter which is chosen in advance, and is just under 1. Frederickson had an important insight which turns out to be central to an understanding of the bin-packing problem--good solutions can be obtained by pairing large elements with small elements.

```
procedure BINPACK; comment from [Fr80];
begin
   place each element which is greater than α in a bin by itself;
   let x_1,x_2,...,x_m be the remaining elements, in increasing order;
   for i := 1 step 1 until ⌊m/2⌋ do
      begin
         if x_i+x_{m-i+1} > 1
            then put x_i and x_{m-i+1} in separate new bins
            else put x_i and x_{m-i+1} together in a new bin;
      end;
   if m is odd then place x_⌈m/2⌉ in a new bin;
end;
```

By a careful choice of α, he is able to cause only a few items to exceed α, and yet guarantee that most of the sums considered in the for-loop are less than 1.

Our algorithm is a slight modification of Frederickson's, which eliminates the need to decide *a priori* on a value for α. (A similar algorithm has been presented by Knödel [Kn81].) For convenience in our later analysis, we will allow the bin capacity to be variable.

```
procedure BINPACK1;
begin
   place the x_i into increasing order;
   lo := 1; hi := n;
   while lo < hi do
      begin
         if x_lo + x_hi is less than the bin capacity
            then
               begin
                  put x_lo and x_hi together in a new bin;
                  lo := lo + 1; hi := hi - 1;
               end
            else
               begin
                  put x_hi in a new bin by itself;
                  hi := hi - 1;
               end;
      end;
   if lo = hi then put x_hi in a new bin by itself;
end;
```

A common problem that arises during the analysis of algorithms with random input is that once the algorithm has run for even a short time, the distribution of the input has been conditioned in a complicated way; fortunately, we can get around this problem for the current analysis by a simple trick. We will let z_i, $i=1,2,...,n+1$ be independent draws from a unit exponential distribution, and set

$$x_0 = 0$$

$$x_{i+1} = x_i + z_{i+1}.$$

Then it will be the case that all the differences between the successive x_i are independent. We will also make a slight change in the statement of the problem: x_{n+1} will be the bin capacity. We will later show how to relate the results obtained under these assumptions to the original distribution with bin capacity 1.

We may now write a revised version of BINPACK1 which gives more insight into the processes involved. SUM will be a variable containing the difference between $x_{lo}+x_{hi}$ and the bin capacity (x_{n+1}). BINS1 (resp. BINS2) will tell the number of bins containing 1 (resp. 2) items. EXP will denote a procedure which returns, at each call, a random variable with an exponential distribution with mean 1. Note that by our definition of the input distribution, decreasing hi by 1 will subtract EXP from SUM, and increasing lo by 1 will add EXP to SUM.

```
procedure BINPACK2;
begin
   SUM := EXP - EXP; BINS1 := BINS2 := 0;
   while at least 2 items remain do
      begin
         if SUM > 0
            then
               begin
                  SUM := SUM - EXP;
                  BINS1 := BINS1 + 1;
               end
            else
               begin
                  SUM := SUM + EXP - EXP;
                  BINS2 := BINS2 + 1;
               end;
      end;
   if one item remains then BINS1 := BINS1 + 1;
end;
```

Be sure to recall that EXP generates an independent drawing at each call; thus EXP-EXP is not identically zero. In fact, a simple calculation establishes the well-known fact that EXP-EXP has the bilateral exponential distribution mentioned in Fact 2.

Now since in the packing produced by this algorithm each bin contains one or two items, it is clear that

$$\text{BINS} = \frac{n + \text{BINS1}}{2}, \tag{1}$$

where BINS is the total number of bins used. We now turn to the analysis of BINS1. Note that as currently written the algorithm is a bit vague, since the *while*-loop involves the condition "at least 2 items remain", and this condition is not explicitly set in the remainder of the algorithm. We could make this explicit by maintaining a count of all EXP values generated during the algorithm; when this count exceeds n, we would know that the variables hi and lo must have met. There is a simpler approach which is sufficient to enable us to obtain a good bound on BINS1. Note that at most n/2 executions of the else clause can occur before all of the items are used up. Moreover, the variable BINS1 is nondecreasing as the algorithm proceeds, so the following algorithm produces a variable B′ such that the expectation of B′+1 is an upper bound on that of BINS1.

```
procedure BINPACK3;
begin
    SUM := EXP - EXP; B' := 0;
    for i := 2 step 2 until n do
        begin
            comment the following loop corresponds to the
                operation of decrementing hi until the
                current x_lo and x_hi fit into a bin;
            while SUM > 0 do
                begin
                    PACK1: SUM := SUM - EXP;
                    if B'<n then B' := B' + 1;
                end;
            comment now we may place x_lo and x_hi
                into a bin together;
            PACK2: SUM := SUM + EXP - EXP;
        end;
end;
```

Let ρ be some real in the range [0,1]. Note that if B′ exceeds some number b, then at least one of the following events must have occurred:

i) At some point, the total of all the quantities added thus far to SUM in statement PACK2 must have exceeded ρb, or

ii) the first b executions of PACK1 must have subtracted less than ρb from SUM.

The probability of (ii) is exponentially small in b by Fact 1. Now let F^{*k} be defined as in Fact 2; then the probability of (i), by that Fact, is at most

$2 - 2F^{*m+1}(\rho b)$, where m is the number of executions of the main loop, namely $\lfloor n/2 \rfloor$. Adding the probability of (i) and (ii),

$$P\{B' \geq b\} \leq 2 - 2F^{*m+1}(\rho b) + (\text{exponentially small terms in } b).$$

Using this inequality, Fact 3, and the fact that F has variance 2, one readily establishes that

$$\begin{aligned}
E[\text{BINS1}] &\leq E[B' + 1] \\
&\leq \int_0^n [(2 - 2F^{*m+1}(\rho b) + (\text{exponentially small terms in } b)]\, db + 1 \\
&= (2/\rho)\ (\tfrac{m}{\pi})^{1/2} + o(m^{1/2}) \\
&= \rho^{-1}\ (\tfrac{2n}{\pi})^{1/2} + o(n^{1/2})
\end{aligned}$$

Since this holds for ρ arbitrarily close to 1, we may conclude that

$$E[\text{BINS1}] \leq (\tfrac{2n}{\pi})^{1/2} + o(n^{1/2}).$$

Thus in view of (1) we obtain

<u>Theorem 1</u>. Under the distribution of input derived above from the exponential distribution, the expected number of bins used by BINPACK1 is at most $n/2+(\frac{n}{2\pi})^{1/2}+o(n^{1/2})$.

<u>Corollary</u>. If we assume each x_i is drawn uniformly and independently from [0,1], and that the bin capacity is 1, the expected number of bins used by algorithm BINPACK1 is at most

$$n/2 + (\tfrac{n}{2\pi})^{1/2} + o(n^{1/2}).$$

(Similar results were obtained by Knödel [Kn81], using Kolmogorov's inequality. His result states that the expected number of bins is

$n/2+O(n^{1/2})$.)

Proof. Note that the behavior of BINPACK1 is completely unaffected if we scale x_i, $1 \leq i \leq n$, and the bin capacity by the same factor. Recall that under the model assumed in Theorem 1, x_{n+1} was used as the bin capacity. Suppose we scale all of the x_i, including x_{n+1}, by dividing by x_{n+1}. Then the bin capacity becomes 1, and by [Fe66, Section III.3, Examples (d) and (e), pp. 74-75] the distribution of $x_1,\ldots,x_n$ becomes exactly that of the order statistics of n uniform independent draws from [0,1]. Thus the behavior of the random variable B' is exactly the same under these two models. ■

3. A lower bound

Here we establish that the result of the previous section is optimal, up to constant factors on the amount of wasted space. In this section, we will again assume that the bin capacity is 1 and that the x_i are n uniform independent draws from [0,1]. Let BINS be a random variable telling the optimum number of bins for a problem instance.

Theorem 2. $E[BINS] \geq n/2 + (\frac{n}{24\pi})^{1/2} (3^{1/2} - 1) + o(n^{1/2})$.

Proof. Let N be a random variable telling the number of items whose weight exceeds 1/2; clearly no two of these can lie in the same bucket, so $BINS \geq N$. Let T be a random variable telling the total of the weights of the items; since each bin has capacity 1, $BINS \geq T$. Now for any two random variables Y and Z, not necessarily independent,

$$P\{\max(Y,Z) \leq x\} \leq \min(P\{Y \leq x\}, P\{Z \leq x\}).$$

Thus

$$P\{BINS \le x\} \le \min(P\{N \le x\},\ P\{T \le x\}).$$

Now let W be a random variable with the PDF

$$F(x) = \begin{cases} P\{T \le x\} & \text{for } x \le n/2 \\ P\{N \le x\} & \text{for } x > n/2 \end{cases}$$

(One easily checks that this is an increasing function.) Then, letting $u^{*n}(x)$ denote the density function for the sum of n uniform draws from [0,1], we have

$$E[BINS] \ge E[W] = n/2 + E[W - n/2]$$

$$\ge n/2 + \int_0^{n/2} (x-n/2)\ u^{*n}(x)\ dx + \sum_{i=\lceil n/2 \rceil}^{n} (i-n/2)\ \binom{n}{i}\ 2^{-n}.$$

Applying Fact 3 to the integral, and Fact 4 to the sum, we obtain

$$\frac{n}{2} - \left(\frac{n}{24\pi}\right)^{1/2} + \left(\frac{n}{8\pi}\right)^{1/2}$$

$$= \frac{n}{2} + \left(\frac{n}{24\pi}\right)^{1/2} (3^{1/2} - 1).$$

■

4. A Bound on the Behavior of Two Common Greedy Algorithms

Two common approximation algorithms for bin packing are best-fit-decreasing (BFD) and first-fit-decreasing (FFD). Each of these algorithms first sorts the items to be packed into order of decreasing size, and then packs them in that order, allocating a new bin only when the item being packed fits in none of the bins currently allocated. If any of the partially allocated bins can hold the item, FFD uses the one which was allocated the earliest, while BFD uses the one which can hold the item with the least leftover space. The following theorem and its proof are quite similar to a corresponding result for the algorithm in [Fr80]. In the theorem, XFD denotes either FFD or BFD.

Theorem 3. The number of bins used by XFD does not exceed the number used by BINPACK1.

Proof. Let A denote the set of bins used in algorithm BINPACK1 which contain an element greater than 1/2. Let B denote the set of bins used by BINPACK1 which contain no element greater than 1/2. Now suppose we run XFD. Note that the elements greater than 1/2 are packed first, and each appears in a separate bin. Thus we may identify these bins with the set A of bins mentioned above for algorithm BINPACK1. Imagine we also give XFD a set B of initially empty bins to use during the packing, of cardinality equal to the set B mentioned above.

Suppose we have partially completed a run of algorithm XFD, and have packed the elements $x_n, x_{n-1}, \ldots, x_{i+1}$ thus far. Assume that $x_i \leq 1/2$. Let $R_i = \{x_i, x_{i-1}, \ldots, x_1\}$ and let S_i denote a multiset of capacities constructed as follows:

a) for each bin b in A which contains only one item, include the remaining capacity of A.

b) for each bin in B which currently contains one element, include the capacity 1/2.

c) for each bin in B which currently is empty, include two copies of the capacity 1/2.

Let M_i denote a pairing of maximum cardinality between items in R_i and capacities in S_i such that no element of R_i or S_i is used more than once, and in each pair the item size is less than or equal to the capacity. We establish the following three facts about the size of these M_i.

A) $|M_h| = h$. To see this, note that the packing used by BINPACK1 packs at most two items to each bin in B, and does not exceed the capacity of any bin in A, so it gives us a matching of cardinality h between the item sizes in R_h and the capacities in S_h.

B) For i=h, h-1, ..., 1, if $|M_i|=i$, then when XFD is packing x_i it can do so without using any bins beyond those provided by the sets A and B. To see this, note that M_i describes a way of packing all of the remaining items into the bins in A and B, so surely there is a way to pack x_i.

C) For i=h, h-1, ..., 2, $|M_{i-1}|\geq|M_i|-1$. Intuitively, the potential problem is that since packing x_i requires us to remove one item from R_i, and can require us to remove one capacity from S_i, the size of the maximum pairing between R_i and S_i could conceivably decrease by 2. Now if x_i is packed into a bin which already had two items, S_{i-1} will be the same as S_i, so this problem does not arise. Suppose that x_i is packed into a bin which contained fewer than two items; let b be the minimum of 1/2 and the remaining capacity of the bin into which x_i is packed by XFD. Then the only case in which the size of the maximum pairing could decrease by more than one is the case in which both x_i and b are used in the pairing M_i, but x_i is not paired with b; assume that this case holds. Let b' be the value paired with x_i, and let x' be the item paired with b. Now since x_i is paired with b', if XFD is BFD we know that b is less than b', for BFD always uses the bin of least possible remaining capacity. If XFD is FFD, we again know that b is less than b', since FFD uses the first feasible bin and the bins corresponding to values in S_i are in order of increasing S_i values. Therefore, whether XFD is BFD or FFD, x′ could be paired with b′, so removing x_i from R_i and b from S_i decreases the cardinality of the maximum pairing by at most 1. Thus $|M_{i-1}| \geq |M_i| - 1$.

From (A), (B), and (C), it follows that XFD will complete the packing without using any bins beyond those provided in sets A and B. ∎

By this theorem and the results of the previous sections, we obtain the following theorem.

Theorem 4. If we assume each x_i is drawn uniformly and independently from [0,1], and that the bin capacity is 1, the expected number of bins used by BFD or FFD is $n/2+\Theta(n^{1/2})$.

5. Other distributions

It would be interesting to investigate the behavior of this problem under other distributions; it appears that the behavior is quite sensitive to changes in the distribution. For example, if the x_i are drawn from a distribution on [0,1] which has mean 1/2 but has a probability of more than 1/2 of being greater than 1/2, it is easy to establish that the expected wasted space in the optimum solution is $\Theta(n)$.

Acknowledgements

I wish to thank a number of people for some very helpful suggestions. Jon Bentley brought this problem to my attention, and made some remarks which were helpful in obtaining the lower bound in Section 3. The contents of Section 4 were suggested to me by Dave Johnson. Scott Huddleston had some useful discussions with me; in particular, when I mentioned the use of partial sums of exponential distributions for introducing independence in a scheduling problem, he suggested that I also use this trick when analyzing bin packing. A number of useful comments were received when an earlier version of this paper was presented at GE Labs in Schenectady. Donald Darling brought references for Facts 1 and 2 to my attention.

Appendix I.

In this Appendix we sketch proofs of some of the technical facts used during the proofs.

Fact 3. Let $f(x)$ be any density function which satisfies the conditions for [Fe66, Chapter XVI.2, Theorem 2, page 508], with $r=4$. Let f have mean zero and variance σ^2, and let f^{*n} denote the pdf for the sum of n independent draws with pdf f. Then for any fixed $\alpha>0$,

$$\int_0^{\alpha n} x\, f^{*n}(x)\, dx = \sigma \left(\frac{n}{2\pi}\right)^{1/2} + o(n^{1/2}). \tag{2}$$

Proof. Define ξ by $x=n^{1/2}\sigma\xi$, where σ^2 is the variance of f (in this case 2). Following [Fe66, Chapter XVI, Section 1, page 505], we define f_n as

$$f_n(\xi) = n^{1/2}\, \sigma\, f^{*n}(n^{1/2}\, \sigma\, \xi).$$

Making a change of variable in the integral of (2) yields

$$\int_0^{\alpha n^{1/2}/\sigma} n^{1/2}\sigma\xi\, f_n(\xi)\, d\xi,$$

which can be written as

$$\int_0^{\alpha n^{1/2}/\sigma} n^{1/2}\sigma\xi\, z(\xi)\, d\xi + \int_0^{\alpha n^{1/2}/\sigma} n^{1/2}\sigma\xi\, (f_n(\xi) - z(\xi))\, d\xi, \tag{3}$$

where $z(\xi)$ denotes the density function for the normal distribution with zero mean and unit variance. Now

$$\int_0^{\alpha n^{1/2}/\sigma} n^{1/2}\sigma\xi\, z(\xi)\, d\xi = \sigma \left(\frac{n}{2\pi}\right)^{1/2} + o(1),$$

by direct calculation. Next, by [Fe66, section XVI.2, Theorem 2, page 508], we have

$$f_n(\xi) - z(\xi) = [n^{-1/2} P_3(\xi) + n^{-1} P_4(\xi)] z(\xi) + o(n^{-1}),$$

where P_3 and P_4 are polynomials whose coefficients do not depend on n. Thus the second integral in (3) can be rewritten as

$$\int_0^{\alpha n^{1/2}/\sigma} n^{1/2}\sigma\xi \, [(n^{-1/2} P_3(\xi) + n^{-1} P_4(\xi)) z(\xi) + o(n^{-1})] \, d\xi$$

$$= \int_0^{\alpha n^{1/2}/\sigma} \sigma \, \xi \, [P_3(\xi) + n^{-1/2} P_4(\xi)] \, z(\xi) \, d\xi + o(n^{1/2}). \tag{4}$$

Now for any polynomial p,

$$\int_{-\infty}^{\infty} |p(\xi)| \, z(\xi) = O(1),$$

so (4) is $o(n^{1/2})$, completing the proof. ■

<u>Fact</u> <u>4</u>. $\displaystyle\sum_{i=\lceil n/2\rceil}^{n} 2^{-n} \binom{n}{i} (i-n/2) = (\tfrac{n}{8\pi})^{1/2} + o(n^{1/2}).$

<u>Proof sketch</u>. Rewrite the sum as

$$\int_{x=\lceil n/2\rceil}^{n+1} 2^{-n} \binom{n}{\lfloor x\rfloor} (\lfloor x\rfloor - n/2) \, dx. \tag{5}$$

Change the lower limit of the integral to n/2; clearly this introduces only an o(1) error. Now let I_1 be the interval $[n/2, n/2+n^{3/5}]$, and I_2 be the interval $[n/2+n^{3/5}, n+1]$. Using [Fe68, VII.3, Theorem 1, page 184], we may rewrite the part of the integral in (5) over I_1 as

$$(1 + o(1)) \, (4/n)^{1/2} \int_{n/2}^{n/2+n^{3/5}} z((4/n)^{1/2}(\lfloor x\rfloor - n/2)) \, (\lfloor x\rfloor - n/2) \, dx$$

$$= (1 + o(1))\ (4/n)^{1/2} \int_{n/2}^{n/2+n^{3/5}} [z((4/n)^{1/2}(x-n/2))\ (x-n/2) + O(1)]\ dx,$$

where z(x) is again the normal density function with zero mean and unit variance, and the equality follows from the fact that the derivative of $x\ z((4/n)^{1/2}x)$ is uniformly bounded. Now by direct computation,

$$(4/n)^{1/2} \int_{n/2}^{n/2+n^{3/5}} z((4/n)^{1/2}(x-n/2))\ (x-n/2)\ dx$$

$$= (\frac{n}{8\pi})^{1/2} + o(1),$$

so the integral of (5) over I_1 is

$$(1 + o(1))\ [(\frac{n}{8\pi})^{1/2} + o(1) + O(n^{3/5}/n^{1/2})]$$

$$= (\frac{n}{8\pi})^{1/2} + o(n^{1/2}).$$

The integral over I_2 may be seen to be o(1), using the above-cited theorem and the monotonicity of $2^{-n}\binom{n}{\lfloor x \rfloor}$ over I_2. Thus the fact follows. ■

References

[Ch52] H. Chernoff, "A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the Sum of Observations," Ann. Math. Stat. 23 (1952), pp. 493-507.

[CT78] Y. S. Chow and H. Teicher, Probability Theory: Independence, Interchangeability, Martingales, Springer-Verlag, New York, 1978.

[CSHY80] E. G. Coffman, Jr., K. So, M. Hofri, and A. C. Yao, "A Stochastic Model of Bin-Packing," Information and Control 44,2 (February 1980), pp. 105-115.

[Fe66] William Feller, An Introduction to Probability Theory and Its Applications, Volume II, Wiley, New York, 1966.

[Fe68] William Feller, An Introduction to Probability Theory and Its Applications, Volume I, 3rd edition, Wiley, New York, 1968.

[Fr80] Greg N. Frederickson, "Probabilistic Analysis for Simple One- and Two- Dimensional Bin Packing Algorithms," Information Processing Letters 11, 4-5 (12 December 1980), pp. 156-161.

[GJ80] David S. Johnson and Michael Garey, "Approximation Algorithms for Bin Packing Problems: A Survey," manuscript, 1980.

[Jo73] David S. Johnson, Near-Optimal Bin Packing Algorithms, Ph.D. Thesis, Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Mass., June 1973.

[Jo74] David S. Johnson, "Fast Algorithms for Bin Packing," Journal of Computer and System Sciences 8 (1974), pp. 272-314.

[JDUGG74] 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. 3,4 (December 1974), pp. 299-325.

[Kn81] W. Knödel, "A Bin Packing Algorithm with Complexity O(n log n) and Performance 1 in the Stochastic Limit," Conference of Mathematical Foundations in Computer Science, Strbske Pleso, Czechoslovakia, 1981.

[Sh77] Stephen D. Shapiro, "Performance of Heuristic Bin Packing Algorithms with Segments of Random Length," Information and Control 35 (1977), pp. 146-158.

[VL81] W. Fernandez de la Vega and G. S. Lueker, "Bin Packing Can Be Solved within 1+ε in Linear Time," Combinatorica 1,4 (1981), to appear.

[Ya80] Andrew C. Yao, "New Algorithms for Bin Packing," J. ACM 27,2 (April 1980), pp. 207-227.