# Asymptotic Probabilistic Analysis of Packing and Related Partitioning Problems

*E. G. Coffman, Jr., D. S. Johnson and P. W. Shor*

Bell Laboratories
AT&T Bell Laboratories
Murray Hill, NJ 07974

## 1 Introduction

**Problems**. The problems studied here all involve the partitioning of a set of positive numbers into a collection of subsets satisfying a sum constraint. The following two problems are among the most fundamental. They have wide-ranging applications throughout computer science and operations research.

*Bin Packing* (BP). Given $c > 0$ and a set $S = \{X_1, \ldots, X_n\}$ with $0 < X_i \leq c$, $1 \leq i \leq n$, partition $S$ into a minimum number of subsets such that the sum of the $X_i$'s in each subset is no more than $c$.

The $X_i$'s are usually called items or pieces and are thought of as being packed into bins $B_1, B_2, \ldots$, each with capacity $c$; the items packed in a bin comprise one of the subsets in a solution to the optimization problem.

*Multiprocessor Scheduling* (MS). Given an integer $m \geq 1$ and a set $S = \{X_1, \ldots, X_n\}$, find a partition of $S$ into $m$ subsets such that among all such partitions, the maximum subset sum is minimized.

Note that the MS problem is complementary to the BP problem in that the objective function and the given parameter are interchanged. The items are now called tasks or jobs, with running times or durations instead of sizes. The bins become processors $P_1, \ldots, P_m$, and the partition becomes a schedule of $S$ on $m$ processors that minimizes the *makespan* $c$, i.e., the completion time of a latest finishing task. Because of the sequential nature of most heuristics, it is convenient to assume that the set to be partitioned is given as a list $L_n = (X_1, \ldots, X_n)$ from which items are packed or scheduled one by one. If $H$ denotes an MS heuristic, then $H(L_n, m)$ denotes the makespan of the $m$-processor schedule generated by $H$ for the tasks in $L_n$. In the BP problem, the bin capacity is essentially a scale factor, so we take $c = 1$ without loss of generality. Thus, if $H$ denotes a BP heuristic, then $H(L_n)$ denotes the number of unit capacity bins in which $H$ packs the items of $L_n$.

Merely deciding whether a list of numbers can be partitioned into two subsets with equal sums is NP-complete, so as one would expect, both the BP and MS problems are NP-complete. Thus, one is unlikely to find an algorithm that will solve these problems exactly and efficiently.[1] For this reason, a large literature has built up over the past 20 years on the design and analysis of heuristic or approximation algorithms. Such algorithms are designed to generate optimal or nearly optimal solutions for most problem instances. Quantifying this last statement is the goal of analysis.

**Analysis**. Early research on BP, MS, and related problems concentrated on combinatorial, worst-case results, as reflected in the survey by Coffman, Garey and Johnson (1984). For example, a scheduling heuristic $H$ would be assessed by determining for each $m$ an upper bound over all $L_n$ and $n$ on the ratio $H(L_n, m)/OPT(L_n, m)$, where OPT stands for an optimal algorithm, i.e., $OPT(L_n, m)$ denotes the makespan of a solution to the MS problem for the problem instance $(L_n, m)$. Similarly, the ratios $H(L_n)/OPT(L_n)$ were bounded for BP heuristics $H$. Such results are inherently pessimistic, so probability models were introduced in order to learn more about the probable or average-case behavior of heuristics. Probabilistic analysis began about 10 years ago, and gained considerable momentum when some striking new results were developed a few years later.

In the standard probability model, the $X_i$'s are taken as independent samples from a given distribution $F(x)$. The goal is then an estimate of distributions such as $P\{H(L_n) \le x\}$, or what is sometimes easier to obtain, expected values such as $E[H(L_n, m)]$, where the expectations are over all $n$-item samples $L_n = (X_1, \ldots, X_n)$.

Typically, exact analysis of probability models is quite difficult, especially for the more efficient algorithms, so asymptotic techniques have been used. These techniques estimate behavior for large problem instances, i.e. for large $n$. Also, the estimates often take the form of expressions with terms that are precise only within unspecified multiplicative constants. For example, let $F(x)$ be the uniform distribution on $[0, 1]$. Then as illustrated later, there are BP heuristics $H$ for which it has been proved that $E[H(L_n)] = n/2 + \Theta(\sqrt{n})$. Here, the $\Theta(\cdot)$ notation is simply a relaxation of the concept "is proportional to." Precisely, $f(n) = \Theta(g(n))$ means that there exist constants $\alpha$, $\beta > 0$ such that for all $n$ large enough,

$$\alpha g(n) \le f(n) \le \beta g(n) \, .$$

If we only know the existence of $\beta > 0$ such that the right-hand inequality is satisfied for all $n$ large enough, then we write the familiar $f(n) = O(g(n))$. A similar restriction to $\alpha$ and the left-hand inequality is denoted $f(n) = \Omega(g(n))$.

We emphasize that usually very little is known about the multiplicative constants hidden in the

[1]Garey and Johnson (1979) give a comprehensive treatment of NP-completeness and its implications.

$\Theta(\cdot)$ terms. One can almost always find some bounds for these constants, but in most cases there is reason to believe that the bounds are very crude.

In the remainder of this section we present a number of fundamental algorithms together with a sampling of results that measure the quality of the packings produced.

**BP results**. We begin by describing three algorithms that pack the items in the sequence $X_1, \ldots, X_n$. An item is packed when it is encountered; once packed, it is not moved thereafter. The algorithms are said to be *on-line* because, for each $i$, $1 \leq i \leq n$, the rule that decides where $X_i$ is packed is independent of the number and sizes of the remaining items $X_{i+1}, \ldots, X_n$. All three algorithms begin by packing $X_1$ into $B_1$.

The simplest of the three algorithms is Next Fit, abbreviated NF. In packing $X_i$, $i \geq 2$, NF first checks the highest indexed, nonempty bin, say $B_j$, $j \geq 1$. $X_i$ is packed in $B_j$ if it fits, i.e., if $X_i$ plus the sum of the items already packed in $B_j$ is at most 1. Otherwise, $X_i$ is packed into $B_{j+1}$, which then becomes the new highest-indexed, nonempty bin.

The two algorithms, First Fit (FF) and Best Fit (BF), improve on NF by checking *all* nonempty bins before starting a new bin, i.e., FF and BF pack an item $X_i$ into an empty bin if and only if $X_i$ does not fit into any nonempty bin. FF packs $X_i$, $i \geq 2$, into the lowest indexed, nonempty bin, if any, in which $X_i$ fits, while BF packs $X_i$ into a nonempty bin, if any, in which $X_i$ fits best, i.e., with the least unused capacity left over. Ties are resolved by BF in favor of lower indexed bins.

Improved, off-line versions of these algorithms are obtained by first sorting the $X_i$'s into decreasing order; the corresponding NFD, FFD, and BFD algorithms (D stands for decreasing) are simply NF, FF, and BF applied to the list $(X_{(n)}, \ldots, X_{(1)})$, where $X_{(i)}$ denotes the $i^{\text{th}}$ smallest item in $L_n$.

Table 1 summarizes a number of the basic results that have been derived for the above algorithms under the assumption that $F(x)$ is the uniform distribution on $[0, 1]$. Worst-case bounds and the average-case result for $OPT(L_n)$ are also shown for comparison. Clearly, the average cases are far more favorable than the worst cases. The ratios $E[H(L_n)]/E[OPT(L_n)] \sim 2E[H(L_n)]/n$ for NF and NFD are 4/3 and 1.29 as compared to their respective worst-case ratios, 2 and 1.691.... Note also that the FF and BF heuristics, along with their counterparts FFD and BFD, are all asymptotically optimal in the sense that $E[H(L_n)]/E[OPT(L_n)] \sim 1$ as $n \rightarrow \infty$ for each heuristic.

The shortcomings of the $\Theta(\cdot)$ results are apparent, since the average-case results do not distinguish between FFD, BFD, and OPT. On the other hand, the average-case results do show that for all $n$ sufficiently large, $FF(L_n) > BF(L_n)$, a distinction that does not appear in the worst-case results.

| Algorithm | l.u.b. for $H(L_n)/OPT(L_n)$ | $E[H(L_n)]$ |
|---|---|---|
| NF | 2 | $\sim 2n/3$ as $n \to \infty$ |
| FF | 17/10 | $= n/2 + \Theta(n^{2/3})$ |
| BF | 17/10 | $= n/2 + \Theta(\sqrt{n} \log^{3/4} n)$ |
| NFD | 1.691 ... | $\sim (.645\ldots)n$ as $n \to \infty$ |
| FFD | 11/9 | $= n/2 + \Theta(\sqrt{n})$ |
| BFD | 11/9 | $= n/2 + \Theta(\sqrt{n})$ |

$$E[OPT(L_n)] \;=\; n/2 + \Theta(\sqrt{n})$$

Table 1. Worst-case vs. average-case with $F(x)$ the uniform distribution on $[0,1]$.

**MS results**. We begin by describing three heuristics. The simplest is the on-line List Scheduling (LS) algorithm, which schedules the tasks in the given sequence $X_1, \ldots, X_n$ on the processors $P_1, \ldots, P_m$ with $X_1$ starting on $P_1$. LS schedules $X_i$, $i \geq 2$, on that processor having a smallest workload in the schedule for $X_1, \ldots, X_{i-1}$, with ties broken in favor of lower indexed processors. By the workload of a processor, we mean the total duration of the tasks already scheduled on that processor. As before, LS can be improved by first sorting $L_n$ into decreasing order. LS along with the initial sorting is called the Largest Processing Time (LPT) algorithm.

The third MS heuristic was originally proposed for a somewhat different optimization problem: With the instances $(L_n, m)$ the same as in the MS problem, the objective of the *set-partitioning* (SP) problem is to find a schedule that minimizes the difference in the maximum and minimum processor workloads. Clearly, one expects a good heuristic for SP to be a good heuristic for MS; indeed, the two problems are obviously identical for $m = 2$. The heuristic described below is a *set-differencing method.* It can be extended to all $m \geq 2$. However, we confine ourselves to the case $m = 2$, since it is easier to describe and analyze.

Two tasks $X$ and $Y$ in list $L$ are said to be *differenced in* $L$ when a new list $L'$ is formed from $L$ by replacing $X$ and $Y$ with a task having duration $|X - Y|$. The Largest-First Differencing (LFD) heuristic applied to $L_n = L_n^{(1)}$ for $m = 2$ begins by differencing the largest two tasks in $L_n^{(1)}$ to form $L_n^{(2)}$. Then the largest two tasks are differenced in $L_n^{(2)}$ to form $L_n^{(3)}$. This procedure continues until a list $L_n^{(n)}$ of one task remains. LFD defines a schedule for $L_n$ by requiring that the tasks differenced in $L_n^{(i)}$, $0 \leq i \leq n-1$, be scheduled on different processors in such a way that the final processor workloads differ by the duration of the task in $L_n^{(n)}$. This schedule is easily developed by working backward through the sequence of differencing operations. First, the task in $L_n^{(n)}$ is put on one or the other of the two processors. Suppose the schedule for $L_n^{(i)}$, $2 \leq i \leq n$, has been formed, and let $X$ and $Y$ be the tasks differenced in $L_n^{(i-1)}$. Then the schedule for $L_n^{(i-1)}$ is formed from the schedule for $L_n^{(i)}$ by removing a task of duration $|X - Y|$, then scheduling $X$ and $Y$ on different processors so as to preserve the processor workload difference, i.e., the duration of the task in $L_n^{(n)}$.

Typical, yet simple illustrations of probabilistic results can be found in the analysis of the "absolute error"

$$A^H(L_n, m) \; = \; H(L_n, m) - OPT(L_n, m) \; . \tag{1.1}$$

Let $m = 2$. Then $E[A^H(L_n, 2) \leq E[D_n^H]/2$, where $D_n^H \equiv D^H(L_n, 2)$ is the difference in the two processor finishing times in the schedule produced by $H$. Now assume that $F(x)$ is the uniform distribution on $[0, 1]$. Then an analysis shows that $E[D_n^{LS}] = 1/3$ and $E[D_n^{LPT}] \leq \frac{e}{2(n+1)}$. A satisfactory analysis of LFD remains an open problem, but for a randomized, more easily analyzed version of LFD, denoted LFD$^*$, it has been shown that there exists a universal constant $c > 0$ such that $E[D_n^{LFD^*}] = O(n^{-c\sqrt{\log n}})$. Thus, while the heuristics have become increasingly more complicated, they have produced increasingly more efficient schedules for large $n$.

## 2 Analytical Techniques

We describe and illustrate below a number of the more important techniques that have been successfully applied in the analysis of BP and MS problems.

### 2.1 Markov Chains

For the simpler BP and MS heuristics, it is sometimes possible to formulate a tractable Markov chain that represents the element-by-element development of partitions. A state of the Markov chain must represent block sums in a suitable way; given the state space, the transition function is defined by the heuristic. Results for general $n$ are obtained by a transient analysis, while asymptotics for large $n$ are obtained by a steady-state analysis.

To illustrate ideas, consider the average-case analysis of LS on $m = 2$ processors, and assume that $F(x)$ is the uniform distribution on $[0, 1]$. Define $V_i$ as the (positive) difference between the processor finishing times after the first $i$ tasks have been scheduled. The following recurrence is easily verified:

$$V_i \; = \; \begin{cases} |V_{i-1} - X_i|, & 1 \leq i \leq n, \\ 0, & i = 0 \; . \end{cases}$$

Since the $X_i$ are i.i.d. random variables, $\{V_i\}_{i \geq 0}$ is a Markov chain. A routine analysis shows that the density for $V_i$ is given by $f_i(x) = 2(1-x)$, for all $i > 2$. Then we obtain the result cited in Section 1, viz. $E[D_n^{LS}] = E[V_n] = 1/3$. Since $OPT(L_n, 2) \geq \sigma(L_n)/2$ and $LS(L_n, 2) = [V_n + \sigma(L_n)]/2$, we also have the relative performance bound

$$\frac{E[LS(L_n,2)]}{E[OPT(L_n,2)]} \le 1 + \frac{E[V_n]}{E[\sigma(L_n)]} = 1 + \frac{2}{3n} .$$

As another example, $\{NF(L_n), l_n\}_{n\ge 1}$ is a bivariate Markov chain, where $l_i$ is the level, i.e., sum of item sizes, in the last bin of an NF packing of $L_i$. An analysis of this chain for $F(x)$ uniform on $[0,1]$ shows that

$$E[NF(L_n)] = \frac{2n}{3} + 6, \quad n \ge 2 ,$$

thus refining the result cited in Table 1. Indeed, an explicit, though complicated expression for the distribution of $NF(L_n)$ can be derived.

Unfortunately, the Markov-chain approach seems to be limited to the relatively simplistic, less efficient heuristics; the state spaces of Markov chains for other heuristics like FF and BF simply become too large and unwieldy.

## 2.2 Bounds

The immediate advantage of bounds is that they lead to a tractable analysis. The sacrifice is that they are limited to providing only partial information. However, this information is often sufficient to choose between alternative heuristics. For example, the results cited in Table 1 for FF, BF, FFD, and BFD were all obtained by bounding techniques, yet they show that for all $n$ sufficiently large, we have $E[FF(L_n)] > E[BF(L_n)] > E[H(L_n)]$, where $H$ stands for either FFD or BFD. As illustrated below, bounding techniques have come in two basic forms.

**Bounding the Objective Function**. In analyzing the BP heuristic $H$, it may be possible to find a function $g(L_n)$ such that $g(L_n) \ge H(L_n)$ for all $L_n$ and such that $E[g(L_n)]$ is easily calculated. Then we have the average-case bound $E[H(L_n)] \le E[g(L_n)]$. A similar assertion applies to the analysis of MS heuristics.

As a concrete example, we consider the LPT heuristic and its absolute error, as defined by (1.1). Since $OPT(L_n,m) \ge \sigma(L_n)/m$, we have

$$\begin{aligned} A^{LPT}(L_n,m) &\le LPT(L_n,m) - \sigma(L_n)/m \\ &\le \max_{1\le i\le n} \left\{ X_{(i)} - \frac{1}{m}\sum_{k=1}^{i} X_{(k)} \right\} . \end{aligned} \tag{2.1}$$

To see the latter inequality, let $i$ be the largest index such that $X_{(i)}$ runs until the end of the schedule. Then just after $X_{(i)}$ is scheduled the average processor idle time up to the end of the schedule is at most $(m-1)X_{(i)}/m \le X_{(i)}$. Each task $X_{(k)}$ scheduled after $X_{(i)}$ reduces the average idle time by $X_{(k)}/m$; (2.1) follows easily.

To illustrate the use of the bound (2.1), we show that

$$A^{LPT}(L_n, m) \rightarrow 0 \text{ (a.s.) as } n \rightarrow \infty , \tag{2.2}$$

when $F(x)$ is strictly increasing in $(0, \delta)$ for some $\delta > 0$, and $E[X_i] < \infty$. Bounding the right-hand side of (2.1) by

$$X_{(\lfloor \epsilon n \rfloor)} + \max\left\{0, X_{(n)} - \frac{1}{m} \sum_{k=1}^{\lfloor \epsilon n \rfloor} X_{(k)}\right\} \ , \quad 0 < F^{-1}(\epsilon) < \delta \ , \tag{2.3}$$

we observe that the first term in (2.3) converges (a.s.) to $F^{-1}(\epsilon)$ as $n \rightarrow \infty$, and that it can be made arbitrarily small by an appropriate choice of $\epsilon$. Also, since $E[X_i] < \infty$, $X_{(n)}/n \rightarrow 0$ (a.s.). Moreover, $\sum_{k=1}^{\lfloor n\epsilon \rfloor} X_{(k)}/n$ converges (a.s.) to a positive constant as $n \rightarrow \infty$ for every $\epsilon > 0$. Thus, the second term within the maximization in (2.3) tends to $-\infty$ (a.s.). We conclude that (2.2) holds.

In another application of (2.1), it has been shown that if $F(x)$ is the uniform distribution on $[0, 1]$, then $E[A^{LPT}(L_n, m)] < c_m \frac{m}{n+1}$, where $c_m$ is a bounded function of $m$ that tends to 1 as $m \rightarrow \infty$.

In some cases, the requirement that a bound hold deterministically for all $L_n$ is too stringent to yield good results. In addition to a bound $H(L_n) \leq g(L_n)$ that always holds, there may exist a sharper bound $g'(L_n)$ such that $H(L_n) \leq g'(L_n)$ except on a set having a small probability $q_n$. If $q_n \rightarrow 0$ sufficiently rapidly that $q_n E[g(L_n)] = o(E[g'(L_n)])$ as $n \rightarrow \infty$, then we have

$$E[H(L_n)] \leq (1 - q_n)E[g'(L_n)] + q_n E[g(L_n)] \sim E[g'(L_n)] \ .$$

**Dominating Algorithms**. A common way to bound $H(L_n)$ is to introduce a simpler, more easily analyzed algorithm $H'$ for which it can be proved that $H'(L_n) \geq H(L_n)$ for all $L_n$. In this case, $H'$ is said to *dominate* $H$. For example, the MATCH packing heuristic iterates the following procedure until all items are packed. Let $S$ denote the set of items that remain to be packed. MATCH first finds a largest item in $S$, say $X$. If $|S| = 1$ or if no remaining item fits with $X$, i.e., $Y + X > 1$ for all $Y \in S - \{X\}$, then MATCH puts $X$ into a bin alone. Otherwise, MATCH puts items $X$ and $X'$ into a bin alone, where $X'$ is a largest remaining item other than $X$ such that $X + X' \leq 1$.

It can be proved without much difficulty that $FFD(L_n) \geq MATCH(L_n)$ and $BFD(L_n) \geq MATCH(L_n)$ for all $L_n$. Moreover, MATCH has the following simple analysis. First, we have $MATCH(L_n) \leq (n+b)/2$, where $b$ is the number of singleton bins in the MATCH packing. The number of singletons with an item no larger than 1/2 is at most one, so $MATCH(L_n) \leq (n+b'+1)/2$ where $b'$ is the number of singletons with an item larger than 1/2. But an inspection of MATCH shows that $b'$ is equal in distribution to $\max\limits_{1 \leq i \leq n} \xi_i$ where $\xi_i$ is a symmetric, $n$-step random walk starting at the origin. Standard results then yield $MATCH(L_n) = n/2 + \Theta(\sqrt{n})$ and hence $H(L_n) = n/2 + O(\sqrt{n})$, where $H$ stands for either FFD or BFD.

## 2.3 Stochastic Planar Matching

Matching problems in one or more dimensions have arisen in the analysis of several packing heuristics. An example in one dimension was given in Section 2.2. Here, we first define a generalization of this matching problem to 2 dimensions and then illustrate how it occurs in the analysis of algorithms.

Let $n$ plus points and $n$ minus points be chosen independently and uniformly at random in the unit square. Let $M_n$ denote a maximum *up-right* matching of plus points to minus points such that if a plus at $(x, y)$ is matched to a minus at $(x', y')$, then $x \leq x'$ and $y \leq y'$. Let $U_n$ denote the number of points left unmatched by $M_n$. The problem of determining the distribution of $U_n$ is called the up-right matching problem. Asymptotic bounds on the expected value are given by

$$E[U_n] \;=\; \Theta(\sqrt{n}\,\log^{3/4} n)\ . \tag{2.4}$$

To illustrate the applications of (2.4), we consider an upper bound analysis of the BF heuristic, assuming that $F(x)$ is the uniform distribution on $[0, 1]$. Define the Modified Best Fit (MBF) heuristic to be the same as BF except that MBF closes a bin to any further items whenever the bin receives an item no larger than 1/2. Clearly, bins in an MBF packing have at most two items. It is not difficult to prove that MBF dominates BF, so that $E[BF(L_n)] \leq E[MBF(L_n)]$.

Next, we describe MBF as a matching procedure. Plot the items of $L_n$ as points in the left half of the unit square so that $X_i$ has a $y$ coordinate $1 - i/n$ and an $x$ coordinate $X_i$ if $X_i \leq 1/2$ and $1 - X_i$ if $1/2 < X_i \leq 1$. $X_i$ is plotted as a plus point if $X_i \leq 1/2$ and as a minus point if $1/2 < X_i \leq 1$. Now match a plus point with a minus point if the corresponding items are placed in the same bin by MBF. By definition of MBF, the minus point must be above the plus point, since the item corresponding to the minus point had to be scanned first. Also, the minus point must be to the right of the plus point, since the two items fit into a single bin. An MBF matching is a maximum up-right matching, as is easily verified. However, the model differs from the original one in two respects. First, points are samples in the left half of the unit square, and second, the $x$ coordinate has been discretized so that $x \in \{0, 1/n, \ldots, (n-1)/n\}$. But (2.4) still holds, since the effects of both differences are limited to changes in the hidden multiplicative constant.

Finally, we observe that $MBF(L_n)$ is the sum of the occupied space $\sigma(L_n)$ and the unoccupied space, the latter quantity being bounded by $U_n$. Thus, $E[MBF(L_n)] = n/2 + \Theta(\sqrt{n}\log^{3/4} n)$ and hence

$$E[BF(L_n)] \;=\; \frac{n}{2} + O(\sqrt{n}\log^{3/4} n) \ .$$

Again using the up-right matching result, a corresponding lower bound can be proved, so we obtain the result for BF cited in Table 1.

## 2.4 Order Statistics

Let $F_n(x)$ denote the empirical distribution function for the sample $L_n = (X_1, \ldots, X_n)$. Probability bounds for the one-sided Kolmogorov-Smirnov statistic, $D^+(L_n) = \sup_x [F_n(x) - F(x)]$, have been used to advantage on a number of occasions when the inequality $D_n^+ \le d$ can be transformed directly into a corresponding bound on a given performance metric. A simple example occurs in the analysis of LS when $F(x)$ is the uniform distribution on $[0,1]$.

Consider the normalized absolute error

$$R^{LS}(L_n, m) \;=\; \frac{LS(L_n, m) - OPT(L_n, m)}{OPT(L_n, m)} \ . \tag{2.5}$$

LS can do no worse than to finish a schedule by running the largest task on some processor while all other processors are idle. It follows that $LS(L_n, m) \le [X_{(n)} + \sigma(L_n)]/m$. This bound together with $OPT(L_n, m) \ge \sigma(L_n)/m$ reduces (2.5) to

$$R^{LS}(L_n, m) \le (m-1)/\sigma(L_n) \ . \tag{2.6}$$

Now $D^+(L_n) \le d$ means that $i/n - X_{(i)} \le d$ simultaneously for all $1 \le i \le n$. A summation then yields $(n+1)/2 - dn \le \sigma(L_n)$, whereupon substitution into (2.6) gives

$$R^{LS}(L_n, m) \le \frac{2(m-1)}{n} \, \frac{1}{1-2d} \ . \tag{2.7}$$

Then (2.7) along with the Smirnov estimate,

$$P\left\{ D^+(L_n) \le \frac{x}{\sqrt{n}} \right\} \;=\; 1 - e^{-2x^2}\left(1 - \frac{2x}{3\sqrt{n}} + O\left(\frac{1}{n}\right)\right)$$

can be used to estimate the distribution of $R^{LS}$.

In contrast to the other, more elaborate applications of Kolmogorov-Smirnov statistics, the simplicity of (2.6) permits a direct approach. In particular, we can use the Chernoff bounds

$$P\{\hat{\sigma}(L_n) \ge x\sqrt{n}\} \le e^{-x^2/2}, \quad P\{\hat{\sigma}(L_n) \le -x\sqrt{n}\} \le e^{-x^2/2}, \tag{2.8}$$

where $\hat{\sigma}(L_n) = \sigma(L_n) - E[\sigma(L_n)]$ and $F(x)$ is any distribution on $[-1,1]$. With $-x\sqrt{n} = -n/4$ in (2.8), we have

$$P\left\{ R^{LS}(L_n, m) > \frac{4(m-1)}{n} \right\} \le P\{\sigma(L_n) - n/2 \le -n/4\} \le e^{-n/32} \ .$$

## 3 Final remarks

Here we mention

- variants such as 2-dimensional packing, dual bin packing
- perfect packing problems
- discrete sets of item sizes
- open problems.

# Asymptotic Probabilistic Analysis of Packing and Related Partitioning Problems

*E. G. Coffman, Jr., D. S. Johnson and P. W. Shor*

Bell Laboratories

AT&T Bell Laboratories

Murray Hill, NJ 07974

*ABSTRACT*