A Self-Organizing Bin Packing Heuristic

JANOS CSIRIK [*] DAVID S. JOHNSON [†] CLAIRE KENYON [‡]
PETER W. SHOR [§] RICHARD R. WEBER [¶]

Abstract

This paper reports experiments with a new and surprisingly robust on-line heuristic for one-dimensional bin packing. This new *Sum of Squares* algorithm (SS) is restricted to the class of "discrete" distributions, i.e., ones in which the bin capacity and all item sizes are rational, as is common in practice. One begins by scaling up the item sizes (and the original unit bin capacity) so as to obtain an equivalent "integral" distribution. One then repeatedly applies the following packing rule: Suppose B is the bin capacity and $n(g)$ is the number of bins in the current packing whose contents total $B-g$, $1 \leq g < B$, i.e., which have a "gap" of size g. Place the next item so as to minimize $\sum_{1 \leq i < B} n(g)^2$.

For all the discrete distributions we have tested, SS appears to produce sublinear expected waste whenever the optimal expected waste is sublinear, something that previously known simple algorithms such as *Best Fit* are unable to do. More precisely, it is known from [CCG91, CCG98] that the optimal expected waste for any such distribution is either $\Theta(n)$, $\Theta(\sqrt{n})$, or $O(1)$, and SS appears to distinguish appropriately between these three cases, although the expected waste for the algorithm may grow as $\Theta(\log n)$ in the third case. The algorithm appears to accomplish this feat by a self-organizing process that eventually favors only those bins that are intermediate steps on the way to the production of perfectly packed bins.

The above claims are supported by extensive experimentation, as well as a newly-discovered approach that enables us to determine the expected behavior of optimal packings for any given discrete distribution. This task was previously observed to be NP-hard in [CCG91, CCG98], but we show how it can be accomplished in time polynomial in the bin capacity B by solving a sequence of linear programs and applying results of [CW90, CCG91, CCG98]. This is technically pseudo-polynomial time, but is quite feasible for bin capacities of 200 or more.

Although SS appears to be essentially optimal when the expected optimal waste is sublinear, it is less impressive when the expected optimal waste is linear. Whereas the expected ratio of the number of bins used by SS to the optimal number appears to go to 1 asymptotically in the first case, we have observed it go as high as 1.5 in the second. Adding special "thresholding" rules to the algorithm can reduce this, but, even better, it appears that a slight tailoring of SS to the distribution in question, based on the variable values in the above LP's, may well suffice to make the ratio go to 1 in all cases.

[*] csirik@inf.u-szeged.hu. Department of Computer Sciences, University of Szeged, Szeged, Hungary.
[†] dsj@research.att.com. AT&T Labs, Room C239, 180 Park Avenue, Florham Park, NJ 07932, USA.
[‡] Claire.Kenyon@lri.fr. Laboratoire de Recherche en Informatique, Bâtiment 490, Université Paris-Sud, 91405 Orsay Cedex, France.
[§] shor@research.att.com. AT&T Labs, Room C239, 180 Park Avenue, Florham Park, NJ 07932, USA.
[¶] rrw1@cam.ac.uk. Statistical Laboratory, University of Cambridge, Cambridge CB2 1SB, England

1 Introduction

In the classical one-dimensional bin packing problem, one is given a list $L = \{a_1, ..., a_n\}$ of items, with a size $s(a_i) \in [0, 1]$ for each item in the list. One desires to find a packing of the items into a minimum number of unit-capacity bins, i.e, a partition of the items into a minimum number of subsets such that the sum of the sizes of the items in each subset is one or less. This problem is NP-hard, so much research has concentrated on designing and analyzing polynomial-time approximation algorithms for it, i.e., algorithms that construct packings that use relatively few bins, although not necessarily the smallest possible number. Of special interest have been *on-line* algorithms, i.e., ones that must permanently assign each item in turn to a bin without knowing anything about the sizes or numbers of additional items, a requirement in many applications.

In this paper we concentrate on the average-case behavior of such algorithms. The key metrics with which we are concerned can be defined using the following notation. For a given algorithm A and list L, let $A(L)$ be the number of bins used when A packs L, let $s(L) = \sum_{a \in L} s(a)$, and let $OPT(L) \geq s(L)$ be the optimal number of bins. For a given probability distribution F on item sizes, let $L_n(F)$ be a random n-item list with item sizes chosen independently according to distribution F. Then the *asymptotic expected performance ratio* for A on F is

$$ER_A^\infty(F) \equiv \limsup_{n \to \infty} \left(E\left[\frac{A(L_n(F))}{OPT(L_n(F))}\right]\right)$$

and the *expected waste rate* for A on D is

$$EW_A^n(F) \equiv E\left[A(L_n(F)) - s(L_n(F))\right]$$

Note that because of the low variance of $s(L_n(F))$ for any fixed F, $EW_A^n(F) = o(n)$ implies $ER_A^\infty(F) = 1$ (although not necessarily vice versa). When the context is clear, we will often omit the "(F)" in the above notation.

To date, the most broadly effective practical on-line bin packing algorithm has been *Best Fit* (BF), in which each item is placed in the fullest bin that currently has room for it. Best Fit has been studied under a significant range of distributions. The classical results concern the continuous uniform distributions $U[0, b]$, where item sizes are uniformly distributed over the real interval $[0, b]$. For $b = 1$ we have $EW_A^n = \Theta(n^{1/2}(\log n)^{3/4})$ [Sho86, LS89], and for $b < 1$ experiments reported in [BJLM83, CCG91] suggest that $ER_A^n > 1$, with a maximum value of approximately 1.014, attained for $b \sim 0.79$.

More recently, the behavior of BF has been studied in [CCG91, CJSW93, KRS98] for the discrete uniform distributions $U\{j, k\}$, $1 \leq j < k$, in which the allowed item sizes are $1/k, 2/k, ..., j/k$, all equally likely. For $k \geq 3$ and $j = k-1$, BF's behavior for $U\{j, k\}$ approximately mimics that for $U[0, 1]$, and we have $EW_A^n = \Theta(n^{1/2}(\log k)^{3/4})$ [CJSW]. Moreover, for $j = k-2$ or $j < \sqrt{2k + 2.25} - 1.5$, much better performance occurs and we have $EW_A^n = O(1)$ [CCG91, KRS98]. However, there appears to exist a constant c such that for k sufficiently large and $c\sqrt{k} < j \leq k - 3$ $ER_A^n > 1$, with the behavior for $U\{j, k\}$ roughly mimicking that for $U[0, j/k]$.

If running time is no object, algorithms with significantly better expected behavior are possible. Rhee and Talagrand [RT93a] have shown that for any fixed distribution F, there is an algorithm X_F such that $ER_{X_F}^\infty(F) = 1$ and such that if $EW_{OPT}^n(F) = o(n)$, then

$EW^n_{X_F}(F) = O(n^{1/2}(\log n)^{3/4})$. Moreover, if one is willing to repeatedly solve instances of a strongly NP-hard partitioning problem as part of the algorithm, this level of asymptotic performance can be attained *without* knowing the distribution F in advance, simply by obtaining better and better estimates of it as one goes along, i.e., by learning F on-line [RT93b].

If one restricts attention to discrete distributions, i.e., ones in which the item sizes are all members of a fixed finite set of rational numbers, even better performance is possible for those distributions F with $EW^n_{OPT} = o(n)$. For discrete distributions F, the only possible values of $EW^n_{OPT}(F)$ are $\Theta(n)$, $\sqrt{n}$, and $O(1)$, as shown in [CCG91, CCG98], and for any fixed discrete distribution F there is a linear time on-line algorithm Y_F that has $EW^n_{Y_F}(F) = O(EW^n_{OPT}(F))$. As was the case with the algorithms X_F, the performance of the algorithms Y_F can also be obtained by a single distribution-free algorithm that learns the distribution as it goes along and repeatedly solves NP-hard problems.

Neither of these generic approaches seems practical, and even the distribution-specific algorithms X_F and Y_F are far too complicated to consider using, requiring the construction of detailed multi-bin packing models that contain slots into which the incoming items must be matched. In this paper we shall present a new and quite simple algorithm *Sum of Squares* (SS) that we conjecture approximately attains the same level of performance as the Y_F for any discrete distribution F, without knowing or attempting to learn F. (We say "approximately" because in some cases where $EW^n_{OPT} = O(1)$, the new algorithm can be shown to yield $EW^n_{SS} = \Omega(\log n)$.) Moreover, although SS like the Y_F's can have $ER^\infty_{SS}(F) > 1$ when $EW^n_{OPT} = \Theta(n)$, there is a simple-to-construct variant SS_F for each such distribution F that we conjecture does yield $ER^\infty_{SS_F}(F) = 1$.

For simplicity in what follows, we shall assume that all discrete distributions have been scaled up by an appropriate multiplier B to obtain an equivalent distribution where item sizes are all integers (and for which the bin capacity is B). For example, the scaled $U\{j,k\}$ distributions have item sizes $1, 2, ..., j$ and bin capacity k. In Section 2, we describe SS and its motivation, and present experimental results comparing it and BF for the distributions $U\{j,k\}$, $1 \le j < k = 100$. It was these results that first suggested to us SS's surprising effectiveness, and led us to an intuitive explanation of its behavior that views the action of the algorithm as a self-organizing process.

For the $U\{j,k\}$ distributions, the needed comparison values of ER^∞_{OPT} and EW^n_{OPT} are already known from theoretical results in [CCG91, CCG98]. For more general classes of discrete distributions, determining these values can be NP-hard. However, as we show in Section 3, the determination can be made by solving a small number of linear programs with $O(B^2)$ variables and $O(B)$ constraints, a process that is feasible for B as large as 200 or more. We use this LP-based approach in Section 4, where we study a generalization of the $U\{j,k\}$ to what we call the *interval distributions* $U\{h{:}j,k\}$, $1 \le h \le j < k$, in which the item sizes, all equally likely, are the integers s, $h \le s \le j$, and the bin capacity is k. We first determine the values of ER^∞_{OPT} and EW^n_{OPT} for all such distributions with $k = 19$ or $k = 100$. Then, based on simulations with 10^5, 10^6 and 10^7 items, we estimate the corresponding values for SS. For $k = 19$ we do this for all relevant values of h and j; for $k = 100$ we do this for a challenging subset of the relevant values. In all cases tested our data is consistent with the hypothesis that $EW^n_{SS} = O(\max\{\log n, EW^n_{OPT}\})$, as claimed. The need for the $\log n$ option is illustrated by tests of the interval distribution $U\{2{:}3,9\}$, and we describe the conditions under which EW^n_{SS} can be proved to grow at least at this rate even though $EW^n_{OPT} = O(1)$.

Finally, in Section 5, we report on various modification of SS aimed at reducing the value of ER^{∞}_{SS} when $EW^n_{OPT} = \Theta(n)$. Although some success can be obtained using generic "bin closing" rules, the most impressive results come when we let ourselves use a small amount of information about the distribution F. In particular, we show how we can use the results of the LP computation we performed to determine the value of $EW^n_{OPT}(F)$ to devise a simple variant SS_F that appears to have $ER^{\infty}_{SS_F} = 1$. This approach can in turn be incorporated into a single "learning" algorithm SS^* that we conjecture yields $ER^{\infty}_{SS^*} = 1$ for all discrete distributions F in time polynomial in n and B.

2 The Sum of Squares Algorithm and $U\{j,k\}$

The sum of squares algorithm works as follows. Assume that our instance has been scaled so that it consists of integer-size items with an integral bin capacity B. Define $n(g)$ to be the number of bins in the current packing whose contents total $B - g$, $1 \le g < B$. Initially $n(g) = 0$, $1 \le g < B$. To pack the next item a_i, we place it in a bin (either a currently empty one or a partially full bin with total contents no more than $B - s(a_i)$) that will yield the minimum updated value of $\sum_{1 \le g < B} n(g)^2$. If there is a tie, we break it in favor of a candidate bin with the largest current total contents.

In proposing this algorithm, our original thought was that it might be good for uniform distributions, since it would tend to maintain an inventory of bins with gaps of all sizes, thus making it likely that a new item would find a bin that it could completely fill. We first tested it on $U\{j,k\}$ distributions, which had been well-studied in the case of Best Fit, and for which the values of EW^n_{OPT} were known from [CCG91, CCG98]. For instance, when $k = 100$, the value we chose for our main tests, $EW^n_{OPT} = O(1)$ for $1 \le j \le 98$ and $EW^n_{OPT} = \sqrt{n}$ for $j = 99$. For each distribution and each $n \in \{10^5, 10^6, 10^7\}$ we computed the average of $SS(L) - s(L)$ and $BF(L) - s(L)$ over a set of random n-item instances to obtain estimates of EW^n_{SS} and EW^n_{BF}. Instances were generated using the "shift register" random number generator described in [Knu81, pages 171–172]. Previous experiments have shown that for bin packing simulations, this choice is unlikely to introduce significant biases.

The results surprised us: EW^n_{SS} appeared to be $O(1)$ for $1 \le j \le 98$, the same range for which $EW^n_{OPT} = O(1)$, and the results for $j = 99$ were consistent with $EW^n_{SS} = O(\sqrt{n})$, again the same value as for EW^n_{OPT}. Table 1 shows our results for $j \in \{24, 25, 60, 97, 98, 99\}$. The first two values of j were chosen as these represent the critical region for Best Fit, where EW^n_{BF} makes a transition from $O(1)$ to $\Theta(n)$. The results for $j = 60$ are typical (except in precise values) of the broad range of j between 25 and 96. The results for $97, 98, 99$ display a critical region for both algorithms, as EW^n_{SS} goes from $O(1)$ to $\Theta(\sqrt{n})$ and EW^n_{BF} goes from $\Theta(n)$ to $O(1)$ to $\Theta(\sqrt{n})$. Our experiments for these last three values of j were extended to include instances with $n = 10^9$, as the rate of convergence is much slower when j is close to k. Indeed, the variance is still sufficiently large for $j = 98$ that we would need substantially more samples if we wanted to get good estimates of the constant to which the expected waste rates are converging.

As suggested by the results in the table, for fixed n the average waste for SS increases monotonically and fairly smoothly with j, but follows a much more adventuresome path for BF. More details on the behavior of BF are reported in [CCG91]. For now it is interesting

Alg	n	Samples	$j = 24$	25	60	97	98	99
SS	10^5	100	223	223	884	23,350	28,510	34,286
	10^6	32	233	249	894	48,896	70,453	105,277
	10^7	10	212	217	797	64,997	150,291	343,958
	10^8	3	267	213	779	82,378	321,068	1,232,118
	10^9	3				68,719	187,061	3,512,397
BF	10^5	100	78	167	16,088	22,669	24,736	25,532
	10^6	32	76	831	154,460	59,015	77,831	88,258
	10^7	10	102	7,737	1,536,747	213,447	185,870	277,278
	10^8	3	67	75,546	15,340,879	1,800,011	254,235	1,081,251
	10^9	3				17,607,786	187,061	2,757,530

Table 1: Measured waste rates for SS and BF under distributions $U\{j, 100\}$.

to note on behalf of Best Fit that although the average waste for BF is enormously larger that that for SS when $25 \leq j \leq 97$ and $EW_{BF}^n = \Theta(n)$, the situation is different when EW_{BF}^n is sublinear, as it is for $1 \leq j \leq 24$ and for $j \in \{98, 99\}$. In these cases its value for fixed n is typically significantly lower than that for EW_{SS}^n, even though the latter has the same growth rate to within a constant factor.

So why does SS do so well in those cases where BF doesn't? Clearly our original idea that it was simply making sure bins were available into which new items would fit exactly does not suffice. For instance, for $U\{25, 100\}$, there are no items available that will fit exactly into gaps of size exceeding 25, even though the algorithm will tend to produce bins with those gaps if none exist. What we now believe is going on is the following. Because of the sum of squares criterion, the creation of bins with a given gap will be inhibited unless there is some way for bins with that gap size to continually disappear. One way for a bin to disappear is for it to have its gap exactly filled; it then no longer contributes to any of the $n(g)$'s. Another way for a bin to disappear, however, is for it to have its gap reduced to one that already disappears for another reason, for instance if the next *two* items it receives will result in exactly filling its gap, or the next three, etc. Thus the algorithm will be driven to favor the creation of precisely those gaps that can (eventually) lead to perfectly packed bins, and the sum of squares criterion is possibly providing a subtle feedback mechanism to maintain the production of the various gaps at the appropriate rates. In other words, it can be thought of as organizing itself for a maximum rate of production of perfectly packed bins. And apparently as long as there exists a scheme that can be expected to pack all but $o(n)$ of its bins perfectly, the algorithm will find it.

3 How to Determine EW_{OPT}^n

In order to test the conjectures made in the previous section, we need a way of determining whether a given discrete distribution F has sublinear $EW_{OPT}^n(F)$. It turns out that this can be formulated as a surprisingly simple linear program based on a network flow model. Suppose our discrete distribution, scaled up to integers, consists of item sizes s_i, $1 \leq i \leq J$, with the probably of s_i occurring being p_i, and let B be the bin size. Our program will have $J(B+1)$ variables $v(i,g)$, $1 \le i \le J$ and $0 \le g \le B$, where $v(i,g)$ represents the rate at which items of size s_i go into bins with gap g. The constraints are:

$$\begin{array}{rcll} v(i,g) & = & 0, & s_i > g \\ \sum_{g=1}^{B} v(i,g) & = & p_i, & 1 \le i \le J \\ \sum_{i=1}^{J} v(i,g) & \le & \sum_{i=1}^{J} \sum_{h=g-s_i} v(i,h), & 1 \le g \le B-1 \end{array}$$

The first set of constraints say that no item can go into a gap that is smaller than it. The second set says that all items must be packed. The third says that bins with a given gap are created at least as fast as they disappear. The goal is to minimize

$$\sum_{g=1}^{B-1} \left(g \left(\sum_{i=1}^{J} \sum_{h=g-s_i} v(i,h) - \sum_{i=1}^{J} v(i,g) \right) \right)$$

Let $c(F)$ be the optimal solution value for the above LP, and let $s(F) = \sum_{i=1}^{J} s_i p_i$ be the average item size under F. Then it can be shown based on results in [CCG91, CW90] that $EW_{OPT}^n = nc(F)/s(F)$ and if $c(F) = 0$, then EW_{OPT}^n is either $\Theta(\sqrt{n})$ or $O(1)$. In the latter case, the determination of which growth rate applies can be made by solving J additional LP's, one for each item size: In the LP for item size s_i, we add an additional variable $x \ge 0$, replace the constraint $\sum_{g=1}^{B} v(i,g) = p_i$ by $\sum_{g=1}^{B} v(i,g) = p_i + x$, add a constraint setting the original objective function to 0, and attempt to maximize x. If the optimal value for x is 0 in any of these LP's, then $EW_{OPT}^n = \Theta(\sqrt{n})$, otherwise it is $O(1)$, again by results in [CCG91, CW90].

Using the software packages `AMPL` and `CPLEX`, we have created an easy-to-use system for generating, solving, and analyzing the solutions of these LP's, given B and a listing of the s_i's and p_i's, or given the parameters h, j, k of an interval distribution. In the next section we describe our results for such distributions.

4 Experiments with General Interval Distributions

In order to test our hypotheses about the performance of SS, we investigated interval distributions $U\{h{:}j,k\}$ for two specific values of k, $k = 19$ and $k = 100$.

For $k = 19$, we considered tested all pairs $h \le j < k$ with $h \le 9$ using the techniques of the previous section to determine ER_{OPT}^∞ and EW_{OPT}^n, and then testing SS and BF on collections of randomly generated instances for the given distribution with $n \in \{10^5, 10^6, 10^7\}$. Pairs h, j with $h \ge 10$ were omitted since for these distribution BF, SS, and OPT all simply place one item per bin and unavoidably have a $\Theta(n)$ expected waste growth. Results are summarized in Table 2.

Note that the "expected" waste rates for OPT in the table are theorems, whereas those for SS and BF are for the most part conjectures with which the data is consistent. (We do

j	Alg	$h=1$	2	3	4	5	6	7	8	9
18	OPT	$\sqrt{n}$	n	n	n	n	n	n	n	n
	SS	$\sqrt{n}$	n	n	n	n	n	n	n	n
	BF	$\sqrt{n}$	n	n	n	n	n	n	n	n
17	OPT	1	$\sqrt{n}$	n	n	n	n	n	n	n
	SS	1	$\sqrt{n}$	n	n	n	n	n	n	n
	BF	1	$\sqrt{n}$	n	n	n	n	n	n	n
16	OPT	1	n	$\sqrt{n}$	n	n	n	n	n	n
	SS	1	n	$\sqrt{n}$	n	n	n	n	n	n
	BF	n	n	$\sqrt{n}$	n	n	n	n	n	n
15	OPT	1	1	n	$\sqrt{n}$	n	n	n	n	n
	SS	1	$\log n$	n	$\sqrt{n}$	n	n	n	n	n
	BF	n	n	n	$\sqrt{n}$	n	n	n	n	n
14	OPT	1	1	$\sqrt{n}$	n	$\sqrt{n}$	n	n	n	n
	SS	1	$\log n$	$\sqrt{n}$	n	$\sqrt{n}$	n	n	n	n
	BF	n	n	n	n	$\sqrt{n}$	n	n	n	n
13	OPT	1	1	1	n	n	$\sqrt{n}$	n	n	n
	SS	1	$\log n$	$\log n$	n	n	$\sqrt{n}$	n	n	n
	BF	n	n	n	n	n	$\sqrt{n}$	n	n	n
12	OPT	1	1	1	n	n	n	$\sqrt{n}$	n	n
	SS	1	$\log n$	$\log n$	n	n	n	$\sqrt{n}$	n	n
	BF	n	n	n	n	n	n	$\sqrt{n}$	n	n
11	OPT	1	1	1	1	n	n	n	$\sqrt{n}$	n
	SS	1	$\log n$	$\log n$	$\log n$	n	n	n	$\sqrt{n}$	n
	BF	n	n	n	n	n	n	n	$\sqrt{n}$	n
10	OPT	1	1	1	1	n	n	n	n	$\sqrt{n}$
	SS	1	$\log n$	$\log n$	$\log n$	n	n	n	n	$\sqrt{n}$
	BF	1	n	n	n	n	n	n	n	$\sqrt{n}$
9	OPT	1	1	1	n	n	n	n	n	n
	SS	1	$\log n$	$\log n$	n	n	n	n	n	n
	BF	1	n	n	n	n	n	n	n	n
8	OPT	1	1	1	1	n	n	n	n	
	SS	1	$\log n$	$\log n$	$\log n$	n	n	n	n	
	BF	1	n	n	n	n	n	n	n	
7	OPT	1	1	1	1	n	n	n		
	SS	1	$\log n$	$\log n$	$\log n$	n	n	n		
	BF	1	n	n	n	n	n	n		
6	OPT	1	1	1	n	n	n			
	SS	1	$\log n$	$\log n$	n	n	n			
	BF	1	n	n	n	n	n			
5	OPT	1	1	1	n	n				
	SS	1	$\log n$	$\log n$	n	n				
	BF	1	n	n	n	n				
4	OPT	1	1	1	n					
	SS	1	$\log n$	$\log n$	n					
	BF	1	n	n	n					
3	OPT	1	1	n						
	SS	1	$\log n$	n						
	BF	1	n	n						
2	OPT	1	n							
	SS	1	n							
	BF	1	n							

Table 2: Orders of magnitude of the measured waste rates under distributions $U\{h{:}j, 19\}$.

n	10^4	10^5	10^6	10^7	10^8	10^9	10^{10}
# Samples	10000	3162	1000	316	100	32	10
Average Waste	7.6	8.6	10.1	10.8	12.1	12.6	14.5
95% Conf. Int.	±0.1	±0.1	±0.2	±0.4	±0.8	±1.0	±1.9

Table 3: Measured average waste for SS under distributions $U\{2{:}3; 9\}$.

have proofs for some of the $h = 1$ entries for BF, in particular those for $j \in \{2, 3, 4, 17, 18\}$ [CCG91, KRS98].) Overall the data is consistent with our conjecture that EW_{SS}^n tracks EW_{OPT}^n when the latter is sublinear. The values of n tested were not sufficiently large for our measurements to make a convincing case for the $\log n$ growth rates reported for SS in the table; in many cases one might just as well have conjectured $EW_{SS}^n = O(1)$. However, the fact that these rates are $\Omega(\log n)$ *is* a theorem.

The intuition behind this theorem is the following. For all the corresponding distributions, there are no items of size 1 but there is at least one item size s that divides $k - 1 = 18$. Thus a sequence of M items of size s, M very large, is likely to create $\Theta(M)$ bins with gap 1 which will never be filled. Such sequences are unlikely for large M, but if one considers a sequence of $(j - h + 1)^M$ items, one can expect such a sequence to occur at least once. This implies that the expected waste must be $\Omega(\log n)$. Typically, the constant of proportionality may be quite small. We can however see this behavior clearly if we consider $U\{2{:}3, 9\}$, a simple distribution with $EW_{OPT}^n = O(1)$. The results for runs of SS on samples of this distribution with n ranging from 10^4 to 10^{10} are summarized in Table 3, and indeed suggest that $EW_{SS}^n \approx \Theta(\log n)$.

For the case of the distributions $U\{h{:}j, 100\}$, Figure 1 displays a graphical representation of the values for EW_{OPT}^n, where an entry of "$-$" represents $O(1)$, an entry of "+" represents $\Theta(\sqrt{n})$, and an entry of "$\cdot$" represents $\Theta(n)$. Note that this picture appears to be a refinement of the structure apparent in Table 2. Moreover, if one ignores the distinction between $-$'s and +'s, it is a fairly accurate discretization of the results for the continuous uniform distributions $U[a, b]$, $0 \leq a \leq b \leq 1$, depicted in Figure 5.2 of [CL91], which partitions the unit square into regions depending on whether $ER_{OPT}^\infty(U[a, b])$ is equal to or greater than 1.

There are far too many $U\{h{:}j, 100\}$ distributions for us to test SS and BF on them all. We therefore have settled for testing isolated examples plus what looks like a challenging slice through Figure 1 – the distributions with $h = 18$, a particularly interesting value because of the large number of transitions that EW_{OPT}^n makes as j goes from h up to $k - h$. In all cases, our experimental results were consistent with the conjecture that $EW_{SS}^n = \Theta(\sqrt{n})$ whenever $EW_{OPT}^n = \Theta(\sqrt{n})$, and $EW_{SS}^n = O(\log n)$ whenever $EW_{OPT}^n = O(1)$.

5 Improving the Performance of SS when $EW_{OPT}^n = \Theta(n)$

Although $ER_{SS}^\infty = 1$ whenever $EW_{OPT}^n = o(n)$, this is not the case when $EW_{OPT}^n = \Theta(n)$. For instance, if $F = U\{34{:}34, 100\}$, i.e., all items have size 34 and $B = 100$, an optimal packing places two items in each bin for a total of $\lceil n/2 \rceil$ bins, whereas SS will create a bin

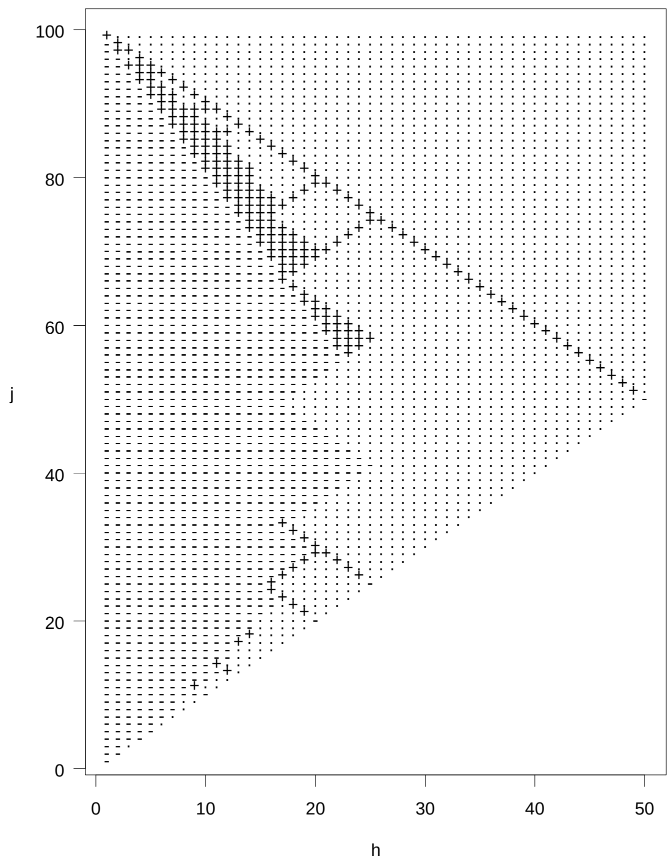

Figure 1: $EW^n_{OPT}(U\{h{:}j, 100\})$: "$-$" means $O(1)$, "+" means $\Theta(\sqrt{n})$, and "$\cdot$" means $\Theta(n)$.

with a single item in it for every two bins that contain two, yielding $ER^\infty_{SS} = 1.2$.

For any bin packing algorithm A, let us define

$$\max ER^\infty_A \equiv \sup\{ER^\infty_A(F) : F \text{ is a discrete distribution}\}$$

Then we have $\max ER^\infty_{SS} \geq 1.2$. In fact, SS can be significantly worse; the performance of SS on the sequence of distributions $U\{2{:}2, 2m+1\}$, $m \to \infty$, implies $\max ER^\infty_{SS} \geq 1.5$. It is thus natural to search for variants on SS that retain its good behavior when $EW^n_{OPT} = o(n)$, while yielding smaller values of $\max ER^\infty_A$. One idea is to add an additional "bin closing" rule to SS. By *closing* a bin we mean declaring it off limits for further items and removing it from the $n(g)$ counts. In SS, the closing rule is simply to close a bin with gap 0, i.e., one that is completely full, as soon as it is created. When $EW^n_{OPT} = \Theta(n)$, even the optimal packing ends up with $\Theta(n)$ incompletely-packed bins, so it might make sense for our algorithm to close some such incompletely-packed bins as well.

We have investigated several closing rules. One fairly effective one is the following two part rule (neither part of which is as good on its own): After one places an item in a bin, close that bin if both (1) the current gap is smaller than the current average gap (over all non-empty bins, open or closed) and (2) the current bin does not have room for a second copy of the item just added to it. Let SS' denote the algorithm that modifies SS by adding this closing rule. This algorithm does substantially better on the distributions that thwarted SS, and the worst distributions we know for it only imply $\max ER^{\infty}_{SS'} \geq 1.1$. Still one would like to do better.

Based on preliminary experiments, it appears that one can indeed do much better, assuming one knows the distribution in advance. It appears that for each discrete distribution F there is a variant SS_F such that $ER^{\infty}_{SS_F}(F) = O(1)$. The modifications to SS involve only the use of a specialized and still fairly simple closing rule.

Suppose we solve the linear program for F presented in Section 3, and let $v^*(i,g)$ be the value of the variable $v(i,g)$ in the optimal solution, $1 \leq i \leq j$ and $0 \leq g \leq B$. For $0 \leq g < B$, define

$$r_g \equiv \sum_{i=1}^{J} \sum_{h=g-s_i} v^*(i,h) - \sum_{i=1}^{J} v^*(i,g)$$

Note that the r_g are non-negative due to the constraints of the LP, and they can be interpreted as the rate at which bins with final gap g are produced in an optimal packing. Our closing rule is the following: When a bin with gap g is created, check to see if the current number of closed bins with gap g is less than nr_g, where n is the number of items in the current packing. If so, close the bin.

Note that when the solution value for the LP is 0 (and hence $EW^n_{OPT} = o(n)$), the only r_g that is positive is r_0, so this rule in a sense generalizes the standard closing rule for SS. The generalization is not exact, however, since when $EW^n_{OPT}(F) = o(n)$, SS_F can occasionally fail to close a bin with gap 0. We should also point out that there are distributions F such that SS_F leaves a significant proportion of the bins with gap 0 open. As an example, consider the distribution with item sizes 25 and 37, where the former appears with probability 1/3 and the latter with probability 2/3. Here an optimal packing will consist almost entirely of bins with two 37's and one 25, and although a bin with gap 0 can be created by putting four 25's together, this will definitely be counterproductive. To inhibit the production of such bins, we must never close them.

Finally, we observe that, assuming that the algorithms SS_F work as well as we claim, there is a pseudopolynomial time distribution-free algorithm SS^* with $ER^{\infty}_{SS^*}F = 1$ for all discrete distributions F. The algorithm works by obtaining better and better estimates of F, based on the items seen so far, and repeatedly re-solving the corresponding LP's to obtain better estimates for the r_g's.

6 Directions of Ongoing Research

At present we are investigating possible ways of obtaining the same behavior claimed for SS^* without actually solving the LP's, as well as investigating some promising approaches toward actually proving the conjectures made in the paper, at least for important special cases. We also are analyzing the worst-case behavior of SS and attempting to obtain tighter bounds on $\max ER^{\infty}_A$ for $A = SS$, SS', and other promising variants. Finally, we note that the Sum of Squares approach should be equally effective for the *bin covering* problem, in which one attempts to maximize the number of bins containing items of total size *at least* B, and we are pursuing this topic, as well as the question of what happens when we are required by our application to keep only a bounded number of bins open.

References

[BJLM83] J. L. Bentley, D. S. Johnson, F. T. Leighton, and C. C. McGeoch. An experimental study of bin packing. In *Proceedings of the 21st Annual Allerton Conference on Communication, Control, and Computing*, pages 51–60, Urbana, 1983. University of Illinois.

[CCG98] E. G. Coffman, Jr., C. Courcoubetis, M. R. Garey, D. S. Johnson, P. W. Shor, R. R. Weber, and M. Yannakakis. Bin packing with discrete item sizes, part I: Perfect packing theorems and the average case behavior of optimal packings. *SIAM J. Disc. Math.* Submitted 1997.

[CCG91] E. G. Coffman, Jr., C. A. Courcoubetis, M. R. Garey, D. S. Johnson, L. A. McGeogh, 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*, pages 230–240. ACM Press, 1991.

[CJSW] E. G. Coffman, Jr., D. S. Johnson, P. W. Shor, and R. R. Weber. Bin packing with discrete item sizes, part IV: Average-case behavior of best fit. In preparation.

[CJSW93] E. G. Coffman, Jr., D. S. Johnson, P. W. Shor, and R. R. Weber. Markov chains, computer proofs, and best fit bin packing. In *Proceedings of the 25th ACM Symposium on the Theory of Computing*, pages 412–421, New York, 1993. ACM Press.

[CL91] E. G. Coffman, Jr. and G. S. Lueker. *Probabilistic Analysis of Packing and Partitioning Algorithms.* Wiley, New York, 1991.

[CW90] C. Courcoubetis and R. R. Weber. Stability of on-line bin packing with random arrivals and long-run average constraints. *Prob. Eng. Inf. Sci.*, 4:447–460, 1990.

[Knu81] D. E. Knuth. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms.* 2nd Edition, Addison-Wesley, Reading, MA, 1981.

[KRS98] C. Kenyon, Y. Rabani, and A. Sinclair. Biased random walks, Lyapunov functions, and stochastic analysis of best fit bin packing. *J. Algorithms*, 27:218–235, 1998.

[LS89] T. Leighton and P. Shor. Tight bounds for minimax grid matching with applications to the average case analysis of algorithms. *Combinatorica*, 9:161–187, 1989.

[RT93a] W. T. Rhee and M. Talagrand. On line bin packing with items of random size. *Math. Oper. Res.*, 18:438–445, 1993.

[RT93b] W. T. Rhee and M. Talagrand. On line bin packing with items of random sizes – II. *SIAM J. Comput.*, 22:1251–1256, 1993.

[Sho86] P. W. Shor. The average-case analysis of some on-line algorithms for bin packing. *Combinatorica*, 6(2):179–200, 1986.