# Average Case Analysis of Bounded Space Bin Packing Algorithms

Nir Menakerman and Raphael Rom

*Department of Electrical Engineering*
*Technion - Israel Institute of Technology*
*Haifa 32000, Israel*
E-mail: mnir@techunix.technion.ac.il

We study the average case performance of bounded space bin packing algorithms. We present a new technique of average case analysis, which is suitable for analyzing a wide variety of algorithms. Our analysis covers algorithms such as Next-$K$ Fit, $K$-Bounded Best Fit and Next Fit Decreasing, as well as of other algorithms. We consider the one-dimensional bin packing problem with discrete item sizes. Discrete item sizes appear in most real-world application of bin packing. However, standard average case analysis assume that items are chosen from a continuous interval. While the continuous distribution is a special case of the discrete distribution, we show that many important results are lost in the transition form the discrete to the continuous distribution. Our technique is general enough to calculate results for any discrete, or continuous, item size distribution. This is significant for real-world applications where the uniform distribution does not always hold.

*Key Words:* bin packing, average case analysis, algorithms, discrete item size distribution, bounded space.

## 1. INTRODUCTION

Because of its applicability to a large number of applications and because of its theoretical interest bin packing has been widely researched and investigated (see, e.g., [12], [17] and [2] for a comprehensive survey). In the classical one-dimensional bin packing problem, we are given a list of items $L = (a_1, a_2, ..., a_n)$, each with a size $a_i \in (0, 1]$ and are asked to pack them into a minimum number of unit capacity bins. Since the problem, as many of its derivatives, is NP-Hard [13], many approximation algorithms have been developed for it (see, e.g., [18], [20], and [1] for a survey). In this paper we restrict our attention to a class of algorithms called bounded space algorithms. An algorithm $A$ is said to be $K - boundedspace$ if at no time during its operation does the number of open bins exceed $K$. This bounded-space restriction arises in many real world applications. For example, consider a communication channel in which variable size datagrams are transmitted in large, fixed-size packets. If the buffer for the channel input is of bounded size, we have a bounded-space bin packing problem. Alternatively, consider the problem of loading containers at a loading dock that has positions for only $K$ containers. If the next item to be packed does not fit in any of the containers, one of them must be closed and shipped out in order to make room for a new container. The analysis of bin packing algorithms is traditionally divided into worst case analysis and average case analysis. In worst case analysis we are usually interested in the ***asymptotic worst case performance ratio***. For a given list of items, $L$ and algorithm $A$, let $A(L)$ be the number of bins used when algorithm $A$ is applied to list $L$, let $OPT(L)$ denote the optimum number of bins for a packing of $L$, and let $R_A(L) \equiv A(L)/OPT(L)$. The asymptotic worst case performance ratio $R_A^\infty$ is defined to be:

$$R_A^\infty \equiv \inf\{r \geq 1 : \text{ for some } N > 0,\ R_A(L) \leq r \text{ for all } L \text{ with } OPT(L) \geq N\} \tag{1}$$

Worst case analysis provides an upper bound on the performance ratio of an algorithm, but from a practical point of view it may be too pessimistic, since the worst case may happen very rarely. A different approach for estimating the performance of an algorithm, is an average case analysis. In this case we assume the items are taken from a given distribution $H$ and we try to estimate the performance ratio of an algorithm, when it is applied to a list taken from that distribution. For a given algorithm $A$ and a list of $n$ items $L_n$, generated according to distribution $H$, the expected performance ratio is defined as follows:

$$\overline{R}_A^n(H) \equiv E\left[R_A(L_n)\right] = E\left[\frac{A(L_n)}{OPT(L_n)}\right] \tag{2}$$

The asymptotic expected performance ratio is defined as:

$$\overline{R}_A^\infty(H) \equiv \lim_{n\to\infty} \overline{R}_A^n(H) \tag{3}$$

Average case analysis enables us to learn more about the typical behavior of the algorithm and provides a better perspective of the worst case analysis. However, the results of an average case analysis depend on the item-size distribution we choose. It is therefore desirable to be able to calculate results for a general distribution. Although other cases have been studied, most of the results that have been computed, concern cases where the items in the list are independent, identically distributed (i.i.d) with a uniform distribution [2]. The uniform distribution appears in two forms:

1. **Continuous Uniform distribution** - denoted by $[a, b]$, where $0 \leq a < b \leq 1$ and the items are chosen uniformly from the continuous interval $[a, b]$.
2. **Discrete Uniform distribution** - denoted by $\{u, U\}$, where $U$ is the bin size and $1 \leq u \leq U$. The items are chosen uniformly from the finite set $\{1, 2, ..., u\}$. Note that for fixed values of $u$ and $U$ the distribution $\{ku, kU\}$ approach the continuous uniform distribution $[0, u/U]$, as $k \to \infty$.

Nearly all previous work on average case analysis of bin packing algorithms assumed a continuous distribution [2]. However in most real-world applications the items are drawn from a finite set. Take for example the communication channel, the size of each datagram is a multiple of some atomic unit (bit, byte etc.) and so is the packet that contains the datagrams. The case of discrete item sizes has been studied by Coffman et al. [3]. In a series of papers, they considered algorithms such as First-Fit [5] and Best-Fit [6] as well as the optimal packing [4]. They showed that the average case behavior of the algorithms can differ considerably between the continuous and discrete distributions. Analytic average case results for bounded space algorithms with discrete item sizes exists only for the Next Fit algorithm [7]. Results for the rest of the algorithms are based on simulations [10].

The main contribution of our work is in presenting an analysis which is suitable for analyzing a wide variety of $K$-bounded space algorithms. Our analysis can be used with any general discrete item size distribution (not necessarily uniform). This is a significant extension to previous work which provided analytic results only for continuous distributions of 1-bounded space algorithms. We present many results that were previously unknown, or were speculated based on simulation. In Section 2 we present the technique of our average case analysis. In Section 3 and Section 4 we analyze the most well known 1-bounded space and 2-bounded space algorithms, respectively. Section 5 explain how the analysis can be extended to higher values of $K$.

## 2. THE AVERAGE CASE ANALYSIS

In this section we present the technique of our average case analysis. We assume some general item size distribution $H$, requiring only that the items be independent, identically distributed. We denote by $h_i$ the probability of an item to be of size $i$, that is, $h_i = Pr\{s(a) = i\}$, $\forall a \in L$. To simplify the presentation we use the Next Fit ($NF$) algorithm as an example. The extension of the technique to other algorithms is straightforward and is presented in the next sections. The $NF$ algorithm keeps only one open bin and packs items, according to their order, into the open bin. When an item does not fit in the open bin, the bin is closed, a new bin is opened and the item is packed in the new bin. We use a Markov chain to describe the packing of the algorithm. The state of the packing, which we denote by $N_t$, is the content of the open bin after $t$ items were packed. Since the bin size is $U$ and there are $n$ items to pack, the possible states of the algorithm are $1 \leq N_t \leq U$, $0 \leq t \leq n$. We consider only the ergodic states of the Markov chain, i.e. the subset of states which are recurrent, aperiodic and accessible from the initial state (empty bins). We elaborate more on the class structure of the Markov chain and its ergodic properties in appendix A, for now let us continue with the analysis.

Assume $N_{t-1} = j$ and the algorithm now packs item $a_t$. If the open bin cannot contain the item, i.e., $j + s(a_t) > U$, the item is packed in a new bin. The previous open bin contains $U - j$ unused units which we call overhead units. We say that the overhead units "increased" the size of $a_t$ and define its combined-size to be the actual size of the item, plus the overhead units it created. For example, say the algorithm is in state $N_t = 2$ and the next item is of size $U$. The overhead in this case is $U - 2$ units and we say the combined size of the item is: $U + U - 2$. Denote by $oh_t$ the overhead added to the size of item $a_t$. For an algorithm $A$ and a list $L_n$ of $n$ items, we define the expected average combined size of all items to be:

$$I_{av}^n(A) \equiv E\left[\frac{1}{n}\sum_{t=1}^{n}(s(a_t) + oh_t)\right] \tag{4}$$

We define the expected *asymptotic average combined size* of all items as:

$$I_{av}(A) \equiv \lim_{n\to\infty} I_{av}^n(A) \tag{5}$$

We can express the asymptotic expected number of bins required by $A$ as:

$$E\left[\lim_{n\to\infty} A(L_n)\right] = \frac{n \cdot I_{av}(A)}{U} \tag{6}$$

We now use a property of the optimal packing that ensures that for any item size distribution the tails of the distribution of $OPT(L_n)$ decline rapidly enough with $n$ [27], so that as $n \to \infty$, $E[A(L_n)/OPT(L_n)]$ and $E[A(L_n)]/E[OPT(L_n)]$ converge to the same limit [9]. Therefore the asymptotic expected performance ratio is given by:

$$\begin{aligned}\overline{R}_A^\infty &= \lim_{n\to\infty} E\left[\frac{A(L_n)}{OPT(L_n)}\right] = E\left[\frac{\frac{U}{n}\lim_{n\to\infty} A(L_n)}{\frac{U}{n}\lim_{n\to\infty} OPT(L_n)}\right] \\ &= \frac{E\left[\frac{U}{n}\lim_{n\to\infty} A(L_n)\right]}{E\left[\frac{U}{n}\lim_{n\to\infty} OPT(L_n)\right]} = \frac{I_{av}(A)}{I_{av}(OPT)}\end{aligned} \tag{7}$$

To find the asymptotic expected performance ratio of the $NF$ algorithm, we must calculate both $I_{av}(OPT)$ and $I_{av}(NF)$. Since bin packing is $NP$-hard, we can not expect to find $I_{av}(OPT)$ for all item size distributions. Fortunately, we do know that for several important distributions, including the uniform distribution, the overhead of the optimal packing can be neglected [2] (see details in subsection 2.1). For such distributions we have:

$$I_{av}(OPT) = \sum_{i=1}^{U} i \cdot h_i \tag{8}$$

To find $I_{av}(NF)$ we use the Markov chain describing the algorithm. Denote by $P$ the transition matrix of the Markov chain and by $\Pi = (\Pi_1, ...\Pi_U)$ the equilibrium probability vector satisfying $\Pi = \Pi P$. Assume $NF$ packs a long list of $n$ items; denote by $n_j$ the number of visits in state $j$ during the packing. Since we consider ergodic chains, we have the following property:

$Pr\left(\lim_{n\to\infty} \frac{n_j}{n} = \Pi_j\right) = 1$, which is usually written as: $\lim_{n\to\infty} \frac{n_j}{n} = \Pi_j$, *a.s. (almost surely).*

We now denote by $n_{j,i}$ the number of items of size $i$ packed when the algorithm is in state $j$. The probability for the next item in the list to be of size $i$, $h_i$, is unrelated to the state of the algorithm. Therefore we can use the Law of large numbers to establish the following property of $n_{j,i}$:

$$\lim_{n\to\infty} \frac{n_{j,i}}{n} = \lim_{n\to\infty} \frac{n_j}{n} \cdot h_i = \Pi_j \cdot h_i, \ a.s. \tag{9}$$

The overhead added to each item is related to both the state of the algorithm and the size of the item. We denote by $oh_i(j)$ the overhead added to an item of size $i$ which is packed when the algorithm is in state $j$. We calculate the average combined size of the items in the following way:

$$\begin{aligned} I_{av}(NF) = \lim_{n\to\infty} I_{av}^n(NF) &= \lim_{n\to\infty} E\left[\frac{1}{n}\sum_{j=1}^{U}\sum_{i=1}^{U} n_{j,i}\cdot(i+oh_i(j))\right] \\ &= E\left[\sum_{j=1}^{U}\sum_{i=1}^{U} \lim_{n\to\infty}\frac{n_{j,i}}{n}\cdot(i+oh_i(j))\right] \end{aligned} \tag{10}$$

Substituting (9) we get:

$$I_{av}(NF) = \sum_{j=1}^{U}\sum_{i=1}^{U} \Pi_j \cdot h_i \cdot (i + oh_i(j)) \tag{11}$$

To simplify (11) we use the following definitions:

$$\begin{aligned} \overline{h} &\equiv \textstyle\sum_{i=1}^{U} i \cdot h_i && \text{average size of items (without overhead)} \\ OH(j) &\equiv \textstyle\sum_{i=1}^{U} h_i \cdot oh_i(j) && \text{average overhead in state } j \\ \overline{OH} &\equiv \textstyle\sum_{j=1}^{U} \Pi_j \cdot OH(j) && \text{average overhead size} \end{aligned} \tag{12}$$

Equation (11) now becomes:

$$\begin{aligned} I_{av}(NF) &= \sum_{j=1}^{U}\Pi_j \cdot \sum_{i=1}^{U} i\cdot h_i + \sum_{j=1}^{U}\Pi_j \cdot \sum_{i=1}^{U} h_i \cdot oh_i(j) \\ &= \overline{h} + \sum_{j=1}^{U}\Pi_j \cdot OH(j) = \overline{h} + \overline{OH} \end{aligned} \tag{13}$$

The expression in (13) is very intuitive; the asymptotic average combined size of the items is the average size of the items plus the average size of the overhead.

In the next subsections we consider how the analysis can be applied to discrete uniform distribution and general distribution.

### 2.1. Discrete Uniform Distribution

The analysis we presented is suitable for any (i.i.d) item size distribution. However, the discrete uniform distribution is of special interest and has been the focus of most previous work [2]. We therefore calculate specific results for the case of discrete uniform distribution. We divide the analysis into two parts:

1. Analysis of the distribution $\{U, U\}$, i.e., $h_i = \frac{1}{U}$, $1 \leq i \leq U$.
2. Analysis of the distribution $\{u, U\}$, i.e., $h_i = \frac{1}{u}$, $1 \leq i \leq u$, $h_i = 0$, $u < i \leq U$.

Obviously when $u = U$ the two distributions are identical. We chose to present a separate analysis of the $\{U, U\}$ distribution since it yields closed form solutions and therefore provides a better understanding of the analysis. Moreover, when $U \to \infty$ we get the continuous uniform distribution [0,1], which has been the focus of most previous work. An important characteristic of the discrete uniform distribution is that the overhead of the optimal packing is negligible. To state it formally, let $L_n$ be a list of $n$ items drawn from distribution $H$. Let $s(L_n)$ be the total size of all items in $L_n$ and define the expected wasted space of algorithm $A$ as:

$$\overline{W}_A^n(H) = E\left[U \cdot A(L_n) - s(L_n)\right] \tag{14}$$

For the discrete uniform distribution $\{u, U\}$, the wasted space of the optimal packing is [3]:

$$\overline{W}_{OPT}^n(\{u, U\}) = \begin{cases} O(1) & u < U - 1 \\ \Theta\left(\sqrt{n}\right) & u \in \{U-1, U\} \end{cases} \tag{15}$$

Based on the above result, we may neglect the overhead of the optimal packing in calculating the asymptotic expected performance ratio.

$$\begin{aligned} I_{av}^n(OPT) &\equiv E\left[\frac{1}{n}\sum_{t=1}^{n}(s(a_t) + oh_t)\right] \\ &= E\left[\frac{1}{n}s(L_n)\right] + \frac{1}{n}\Theta\left(\sqrt{n}\right) \quad \overset{n\to\infty}{\longrightarrow} \quad E\left[\frac{1}{n}s(L_n)\right] \end{aligned} \tag{16}$$

We conclude that for the distribution $\{u, U\}$, the average combined size of the items for the optimal packing is:

$$I_{av}(OPT) = \sum_{i=1}^{u} i \cdot h_i = \frac{1}{u}\sum_{i=1}^{u} i = \frac{u+1}{2} \tag{17}$$

Another important characteristic of the $\{u, U\}$ distribution concerns the class structure of the Markov chain describing the state of the packing. We show, in Appendix A, that the Markov chain has only one subset of recurrent states. Furthermore, excluding the case of $u = 1$, all recurrent states are aperiodic which means they are ergodic. This means that the equilibrium probabilities $\Pi = (\Pi_1, ..., \Pi_U)$ exist and are independent of the initial state of the packing. We find the equilibrium probabilities by constructing the transition matrix $P$ and solving the set of equations defined by: $\Pi = \Pi P$. In some (simple) cases it is possible to obtain a closed form solution of the equilibrium probabilities. In cases where this is not possible, we find the equilibrium probabilities by numerical analysis.

In the next sections we use the discrete uniform distribution to calculate the asymptotic expected performance ratio of several algorithms. Since the average combined size of the optimal packing is known, our objective is to find the average combined size of the items for each algorithm we study.

### 2.2. General Item Size Distribution

Recall that for a general distribution we denote by $h_i$ the probability of an item to be of size $i$. Our only assumption is that the items are i.i.d. The key problem in deriving the expected performance ratio is that, unlike the uniform distribution, $I_{av}(OPT)$ is not known. The decision problem of bin packing is NP-complete, which means we cannot hope to find $I_{av}(OPT)$ in polynomial time for all distributions, as this would imply we can also find $OPT(L)$ in polynomial time.

In cases where $I_{av}(OPT)$ is not known we calculate the ratio between the number of bins used by the algorithm, $A(L)$ and $s(L)/U$. We call this ratio the utilization factor of the algorithm. The utilization factor equals the inverse of the average content of the bins packed by $A$. Note that the utilization factor serves as an upper bound on the performance ratio, since $s(L)/U \leq OPT(L)$. For example, if all items are of size $3U/4$ the average combined size (of any algorithm) is $U$ and the utilization factor is $4/3$. However in this case the performance ratio is 1, since the optimal packing produces the same packing.

We use (13) to calculate the average combined size of the items. We must find two components:

1. The equilibrium probabilities of the Markov chain, $\Pi$.
2. The overhead component $oh_i(j)$, i.e., the overhead added to an item of size $i$ which is packed when the algorithm is in state $j$.

To calculate the equilibrium probabilities we first construct the transition matrix $P$. Once we have the transition matrix, we find the equilibrium probability vector by solving the equation $\Pi = \Pi P$. The calculation of the overhead component $oh_i(j)$, for all the algorithms we consider, is also simple. Hence calculating the average combined size of the items is straightforward. However, since we rely on numerical computations, when the number of states grows the computation becomes harder in terms of time and memory requirements. We present the analysis of a general item size distribution only for the $NF$ algorithm (see, 3.1.3), the analysis of the other algorithms is similar.

## 3. ANALYSIS OF 1-BOUNDED SPACE ALGORITHMS

In this section we consider algorithms that keep only one open bin. The algorithms we consider are Next Fit, Smart Next Fit and Next Fit Decreasing. We present the analysis of the three algorithm and then summarize the results in subsection 3.4. Note that while Next Fit and Smart Next Fit are online algorithm with $O(n)$ running time, Next Fit Decreasing is an off line algorithm with $O(n\, log\, n)$ running time.

### 3.1. The Next Fit Algorithm

The Next-Fit ($NF$) algorithm is perhaps the simplest algorithm for bin packing and one of the first to be studied. The first average case analysis of the $NF$ algorithm was reported by Coffman, So, Hofri and Yao [8], who showed that the asymptotic expected performance ratio for the continuous uniform distribution $[0, 1]$ is: $\overline{R}^{\infty}_{NF} = \frac{4}{3}$. Results for the $[0, b]$ distribution, where $0 < b \leq 1$, have been reported by Karmarkar in [16]. The only results for discrete item sizes are for the $\{U, U\}$ distribution. It has been shown in [7] that the $NF$ algorithm has the following asymptotic expected performance ratio:

$$\overline{R}^{\infty}_{NF}(\{U, U\}) = \frac{2(2U + 1)}{3(U + 1)} \tag{18}$$

Note that the result for the continuous uniform distribution is reached when $U \to \infty$. The above mentioned results were achieved by using different techniques, all of which are fairly complicated (see, for example [8], [16] and [14]). We show how the same results can be obtained using the average case analysis we presented in the previous section.

*3.1.1. The $\{U,U\}$ Distribution*

To calculate the combined average size of the items, we first find the equilibrium probabilities of the Markov chain. There is a symmetry in the lines of the transition matrix $P$, in a sense that line $j$ and line $U-j$ are identical. For $j \le \lfloor \frac{U}{2} \rfloor$ we have:

$$P_{j,k} = \frac{1}{U} \cdot \begin{cases} 0 & 1 \le k \le j \\ 1 & j < k \le U-j \\ 2 & U-j < k \le U \end{cases} \quad \Bigg| \quad 1 \le j \le \left\lfloor \frac{U}{2} \right\rfloor \tag{19}$$

The last line is: $P_{U,k} = \frac{1}{U}, \quad 1 \le k \le U$
The simple structure of the matrix $P$ enables an easy solution to the set of equations $\Pi = \Pi P$:

$$\Pi_j = Pr\left(N=j\right) = \frac{2j}{U\left(U+1\right)} \tag{20}$$

Next we compute the overhead component $OH(j)$. It is easy to verify that for the $NF$ algorithm the average overhead in state $N=j$ is:

$$OH(j) = \sum_{i=1}^{U} h_i \cdot oh_i(j) = \sum_{i=U-j+1}^{U} \frac{1}{U} \cdot (U-j) = \frac{j(U-j)}{U} \tag{21}$$

We now use (20) and (21) to find the average combined size of the items:

$$\begin{aligned} I_{av}(NF) &= \frac{U+1}{2} + \sum_{j=1}^{U} \Pi_j \cdot OH(j) = \frac{U+1}{2} + \sum_{j=1}^{U} \frac{2j}{U(U+1)} \cdot \frac{j(U-j)}{U} \\ &= \frac{U+1}{2} + \sum_{j=1}^{U} \frac{2j^2\left(U-j\right)}{U^2\left(U+1\right)} = \frac{U+1}{2} + \frac{U-1}{6} = \frac{2U+1}{3} \end{aligned} \tag{22}$$

We use $I_{av}(NF)$ and $I_{av}(OPT)$ to obtain the asymptotic expected performance ratio:

$$\overline{R}_{NF}^{\infty}(\{U,U\}) = \frac{I_{av}(NF)}{I_{av}(OPT)} = \frac{(2U+1)/3}{(U+1)/2} = \frac{2(2U+1)}{3(U+1)} \tag{23}$$

The result for the asymptotic expected performance ratio is in accordance with known results reported in [7].

*3.1.2. The $\{u,U\}$ Distribution*

To calculate the transition probabilities we assume that at time $t-1$ the algorithm is in state $N_{t-1} = j$ and the next item to be packed is of size $i$, $1 \le i \le u$. We distinguish between two cases:

1. $j+i \le U$ : In this case the item fits in the open bin. Therefore the next state is $N_t = j+i$.
2. $j+i > U$ : In this case the item does not fit in the open bin, therefore a new bin is opened and the next state is $N_t = i$.

It is now straightforward to construct the transition matrix $P$. Once we have $P$, we can find the equilibrium probability vector satisfying: $\Pi = \Pi P$. We could not find a closed form solution for the set of equations for all values of $u$. However calculating $\Pi$ numerically is easy. Our next step is to calculate $OH(j)$, the average overhead in state $N=j$. Assume that the next item to be packed is of size $i$, $1 \le i \le u$. Note that overhead units are added only if the next item does not fit in the open bin, that is, $i > U - j$. When overhead units are added the overhead is the unused space, which is $U - j$. For the distribution $\{u, U\}$ the average overhead is therefore:

$$OH(j) = \begin{cases} 0 & j \le U - u \\ \frac{(j+u-U)\cdot(U-j)}{u} & j > U - u \end{cases} \tag{24}$$

Now that we have the equilibrium probabilities $\Pi_j$ and the average overhead in state $j$, $OH(j)$, we use (13) to find the average combined size of the items, $I_{av}(NF)$. The asymptotic expected performance ratio is calculated from the following expression:

$$\begin{aligned} \overline{R}^{\infty}_{NF}(\{u, U\}) &= \frac{I_{av}(NF)}{I_{av}(OPT)} = 1 + \frac{2}{u+1}\sum_{j=1}^{U} \Pi_j \cdot OH(j) \\ &= 1 + \frac{2}{u+1}\sum_{j=U-u+1}^{U} \Pi_j \cdot \frac{(j+u-U)\,(U-j)}{u} \end{aligned} \tag{25}$$

If we take $U \to \infty$ we approach the continuous uniform distribution $[0, b]$, where $b = u/U$. Our results for the continuous case, match the results reported in [16]. We present some computational results for the $NF$ algorithm in Section 3.4.

*3.1.3. General Item Size Distribution*

In this section we demonstrate how the analysis can be applied to any item size distribution. We assume the items are i.i.d and the probability to draw an item of size $i$ is $h_i$. As we mentioned earlier, since $OPT(L)$ is not known, we calculate the utilization factor which is the ratio between the number of bins used by the algorithm, $NF(L)$ and $s(L)/U$. We use (13) to calculate the average combined size of the items. The construction of the transition matrix and the calculation of the equilibrium probabilities is similar to the one presented in the previous section. The calculation of the overhead component $oh_i(j)$ (overhead added to an item of size $i$ packed in state $j$) is simple:

$$oh_i(j) = \begin{cases} 0 & j + i \le U \\ U - j & j + i > U \end{cases} \tag{26}$$

**Example**: Consider the case of a communication channel in which variable size datagrams are transmitted in fixed-size packets. We assume common Ethernet datagrams sizes and probabilities; where the packet size is 1024 bytes and the datagrams are of sizes 64, 128, 256 and 1024 bytes with probabilities 0.6, 0.1, 0.05 and 0.25, respectively. To perform the calculation we set $U = 1024$, $h_{64} = 0.6$, $h_{128} = 0.1$, $h_{256} = 0.05$ and $h_{1024} = 0.25$ (in this example we can scale the problem by dividing all sizes by 64). Using our average case analysis we find that the utilization factor is: $\overline{R}^{\infty}_{NF}(H) = 1.423$, hence the channel utilization is: $S = \left(\overline{R}^{\infty}_{NF}(H)\right)^{-1} = 0.702$. It is easy to verify that in this case the performance ratio and the utilization factor are equal. It is interesting to note that the performance ratio is considerably higher than that of the continuous uniform distribution $[0, 1]$, which is 1.333. The example illustrates the importance of being able to calculate the performance ratio for a general distribution, since using the results of a uniform distribution may be misleading.

### 3.2. The Smart Next Fit Algorithm

The smart Next Fit ($SNF$) algorithm has been devised and analyzed by Ramanan [25]. The algorithm is obtained by slightly modifying the Next Fit algorithm. Assume that the level (sum of packed items) of the current open bin, $B_j$, is $c$ and the next item to be packed is of size $i$. If the item does not fit in the open bin, it is packed in a new bin, $B_{j+1}$. The $NF$ algorithm always closes $B_j$ and $B_{j+1}$ becomes the open bin. The $SNF$ algorithm closes the bin with the higher level (ties broken in favor of $B_j$), i.e., if $c < i$ the next item is packed in a new bin which is immediately closed, and $B_j$ remains the open bin.

The $SNF$ algorithms lies somewhere between a 1-bounded space and a 2-bounded space algorithm. We present it as a 1-bounded space algorithm because the state of the algorithm can be described as the content of only one bin.

### *3.2.1. The $\{U, U\}$ Distribution*

The analysis of $SNF$ is similar to that of $NF$. To simplify the equations we assume throughout the analysis that $U$ is even and therefore $U/2$ is an integer. To handle odd values of $U$ it is necessary to replace $\frac{U}{2}$ by $\lfloor \frac{U}{2} \rfloor$ in all the equations.

We use (13) to find the average combined size of the items. The transition matrix $P$ is given by:

$$P_{j,k} = \frac{1}{U} \cdot \left\{ \begin{array}{cl} 0 & 2 \le j \le U/2\,,\ k \le j-1 \\ 0 & U/2+1 \le j < U\,,\ k \le U-j \\ j & j = k\,,\ j \le U/2 \\ U+1-j & j = k\,,\ U/2+1 \le j \le U \\ 1 & else \end{array} \right| \quad 1 \le j \le U \tag{27}$$

Solving the set of equations $\Pi = \Pi\, P$, we obtain the following expression for the equilibrium probabilities:

$$\Pi_j = \left\{ \begin{array}{cl} \frac{j}{(U-j)(U-j+1)} & 1 \le j \le U/2 \\ \frac{1}{j} & U/2+1 \le j \le U \end{array} \right. \tag{28}$$

We now find the overhead component $OH(j)$. Assume the algorithm is in state $j$ and the size of the next item is $i$. We distinguish between two cases - based on the state:

1. The state is $j \le U/2$. In this case an item of size $i > U-j$ is packed in a new bin, which is immediately closed, therefore $oh_i(j) = U - i$. An items of size $i \le U - j$ is packed in the open bin without overhead.

$$OH(j) = \sum_{i=1}^{U} \frac{1}{U} \cdot oh_i(j) = \frac{1}{U} \sum_{i=U-j+1}^{U} (U-i) = \frac{j(j-1)}{2U}\,, \quad 1 \le j \le U/2 \tag{29}$$

2. The state is $j > U/2$. In this case an item of size $j < i \le U$ is packed in a new bin, which is immediately closed, therefore $oh_i(j) = U - i$. An item of size $U - j < i \le j$ is packed in a new bin, which becomes the open bin, therefore $oh_i(j) = U - j$. All other items are packed in the open bin without overhead.

$$\begin{aligned} OH(j) &= \frac{1}{U} \sum_{i=U-j+1}^{j} (U-j) + \frac{1}{U} \sum_{i=j+1}^{U} (U-i) \\ &= \frac{(U-j)(3j-U-1)}{2U}\,, \quad u/2 < j \le U \end{aligned} \tag{30}$$

We can now calculate the combined average size of the items:

$$\begin{aligned} I_{av}(SNF) &= \frac{U+1}{2} + \sum_{j=1}^{U/2} \frac{j}{(U-j)(U-j+1)} \cdot \frac{j(j-1)}{2U} \\ &+ \sum_{j=U/2+1}^{U} \frac{1}{j} \cdot \frac{(U-j)(3j-U-1)}{2U} \end{aligned} \tag{31}$$

Using (31) we find the asymptotic expected performance ratio of $SNF$. We present some computational results for the $SNF$ algorithm in Section 3.4. When $U \to \infty$ we get the uniform continuous distribution. Our result match the one reported in [25]: $\overline{R}^{\infty}_{SNF}([0,1]) = 1.227...$.

*3.2.2. The $\{u, U\}$ Distribution*

To calculate the transition probabilities we assume that at time $t-1$ the algorithm is in state $N_{t-1} = j$ and the next item to be packed is of size $i$, $1 \le i \le u$. We distinguish among three cases:

1. $j + i \le U$: In this case the item fits in the open bin and the next state is $N_t = j + i$.
2. $j + i > U$, $j \ge i$: The item does not fit in the open bin, it is packed in a new bin and the old bin is closed, the next state is $N_t = i$.
3. $j + i > U$, $j < i$: The item does not fit in the open bin, it is packed in a new bin which is immediately closed. The open bin is not changed and the next state is $N_t = j$.

Based on the above rules we can construct the transition matrix $P$ and find the equilibrium probabilities. We could not find a closed form solution for the probabilities for all values of $u$, therefore, we calculate $\Pi$ numerically.

Our next step is to calculate $OH(j)$, the average overhead in state $N = j$. Assume that the next item to be packed is of size $i$, $1 \le i \le u$. Note that overhead units are added only if the next item does not fit in the open bin, that is, $i > U - j$. When overhead units are added, there are two possibilities: If $i \le j$ the open bin is changed and the overhead is $U - j$. If $i > j$ the item is packed in a new bin which is immediately closed and the overhead is $U - i$.

$$oh_i(j) = \begin{cases} 0 & j + i \le U \\ U - j & j + i > U,\ j \ge i \\ U - i & j + i > U,\ j < i \end{cases} \tag{32}$$

For the distribution $\{u, U\}$ the average overhead is therefore (again we use $U/2 = \lfloor U/2 \rfloor$):

$$OH(j) = \begin{cases} 0 & j \le U - u \\ \sum_{i=U-j+1}^{u} (U - i) & U - u < j \le U/2 \\ (min\{u, j\} - U + j)(U - j) + \sum_{i=j+1}^{u} (U - i) & max\{U - u,\ U/2\} < j \end{cases} \tag{33}$$

Now that we have the equilibrium probabilities $\Pi$ and the average overhead in state $j$, $OH(j)$, we use (13) to find the average combined size of the items, $I_{av}(SNF)$. The asymptotic expected performance ratio is calculated from the following expression:

$$\overline{R}^{\infty}_{SNF}(\{u, U\}) = \frac{I_{av}(SNF)}{I_{av}(OPT)} = 1 + \frac{2}{u+1} \sum_{j=1}^{U} \Pi_j \cdot OH(j) \tag{34}$$

The results for the continuous case, i.e., when $U \to \infty$, match the results reported in [25].

Calculating results for a general distribution is done in a similar way. We construct the transition matrix to calculate the equilibrium probabilities and use (32) as the overhead component.

### 3.3. The Next Fit Decreasing Algorithm

The Next Fit Decreasing ($NFD$) algorithm is different from all other algorithms we consider in this paper, since it is an offline algorithm. The algorithm first orders the items in decreasing (non increasing) order and then applies the Next Fit algorithm on the sorted list. As we shell see the analysis of the algorithm is also different from the analysis of the other algorithms. We note that the analysis of the Next Fit Increasing (NFI) algorithm, which packs the items in increasing (non decreasing) order, is identical.

Various methods of probabilistic analysis of the $NFD$ algorithm, for the continuous uniform distribution [0,1] have been presented in [11],[15] and [26]. It has been shown that the asymptotic expected performance ratio of the algorithm is: $\overline{R}^{\infty}_{NFD}([0,1]) = 2\left(\frac{\pi^2}{6} - 1\right) = 1.289...$

#### 3.3.1. *The $\{U,U\}$ Distribution*

We are interested in the average combined size of the items. However, we do not use a Markov chain in the calculation. Instead we look at each size $1 \leq i \leq U$ and find the average combined size of items of size $i$.

To simplify the equations we assume that $U$ is even, for an odd value of $U$ it is necessary to replace $U/2$ by $\lfloor U/2 \rfloor$ in all equations. We observe the following:

1. Items of size $i > U/2$ are packed one in a bin. Therefore their combined size is $U$.
2. The average combined size of items of size $i \leq U/2$ is: $U/\lfloor U/i \rfloor$.

The asymptotic average combined size of the items is:

$$\begin{aligned} I_{av}(NFD) &= \sum_{i=1}^{U/2} \frac{1}{U} \cdot \frac{U}{\lfloor U/i \rfloor} + \sum_{i=U/2+1}^{U} \frac{1}{U} \cdot U \\ &= \sum_{i=1}^{U/2} \left(\lfloor U/i \rfloor\right)^{-1} + \frac{U}{2} \end{aligned} \tag{35}$$

The average combined size of the optimal packing is: $I_{av}(OPT) = \frac{U+1}{2}$, and the asymptotic expected performance ratio is:

$$\overline{R}^{\infty}_{NFD}(\{U,U\}) = \frac{U}{U+1} + \frac{2}{U+1} \sum_{i=1}^{U/2} \left(\lfloor U/i \rfloor\right)^{-1} \tag{36}$$

When $U \to \infty$ we get the asymptotic expected performance ratio of $NFD$, for the uniform continuous distribution [0,1]. Our result is in agreement with the one reported in [11].

The results of the analysis of $NFD$ are quite surprising. The expected performance ratio of the algorithm has a unique (oscillating) behavior with a strong dependence on the value of $U$, the bin size (see Figure 1). Unlike $NF$ and $SNF$ the ratio is not monotonically increasing with $U$. Such combinatorial characteristics can only be revealed by a discrete item size analysis, the continuous analysis can only indicate the asymptotic value.

#### 3.3.2. *The $\{u,U\}$ Distribution*

Extending the previous analysis to the $\{u, U\}$ distribution is straightforward. The only difference is that we must stop the summation at $u$ instead of $U$. Thus (35) becomes:

$$I_{av}(NFD) = \sum_{i=1}^{min\{U/2,\, u\}} \frac{1}{u} \cdot \frac{U}{\lfloor U/i \rfloor} + \sum_{i=U/2+1}^{u} \frac{1}{u} \cdot U \tag{37}$$

Note that the second term in (37) is zero for any $u \leq U/2$.

The extension of the analysis to a general distribution is immediate. We should only change (35) to give proper weight to each item size.

$$I_{av}(NFD) = \sum_{i=1}^{U/2} h_i \cdot \frac{U}{\lfloor U/i \rfloor} + \sum_{i=U/2+1}^{U} h_i \cdot U \tag{38}$$

### 3.4. Summary of Results

We present some results of our average case analysis, for several values of $U$, in Table 1.

**TABLE 1.**

Asymptotic expected performance ratio for distribution $\{U, U\}$.

| U= | 2 | 3 | 4 | 5 | 10 | 100 | $\infty$ |
|---|---|---|---|---|---|---|---|
| $\overline{R}_{NF}^{\infty}$ | 1.1111 | 1.1667 | 1.2 | 1.2222 | 1.2727 | 1.3267 | 1.3333 |
| $\overline{R}_{SNF}^{\infty}$ | 1 | 1.0833 | 1.1 | 1.1278 | 1.1712 | 1.2213 | 1.2274 |
| $\overline{R}_{NFD}^{\infty}$ | 1 | 1.1667 | 1.1 | 1.2333 | 1.2061 | 1.2798 | 1.2899 |

In Figure 1 we present the expected performance ratio for the $\{U, U\}$ distributions for all values of $U \leq 100$. Note that the expected performance ratio of $NF$ and $SNF$ is monotonic increasing with $U$ and the difference between their ratios is almost constant. The $NFD$ algorithm, on the other hand, has a totally different behavior, the performance ratio is oscillating but has an asymptotic limit.

The graph in Figure 2 presents the expected performance ratio of the three algorithms for the $\{u, U\}$ distributions, when $U = 100$ and values of $u \leq 100$. The pattern of graphs for other values of $U$, including the case of $U \to \infty$, is similar with a small displacement in the Y axis. As we expect, the performance ratio of $NF$ and $SNF$ is identical for all values of $u \leq U/2$. The performance ratio of $NFD$ is oscillating. It is interesting to note that the performance ratio of $NFD$ is worse than that of $NF$ for a wide range of values of $u$.

## 4. ANALYSIS OF 2-BOUNDED SPACE ALGORITHMS

In this section we analyze several algorithms that use two open bins. The algorithms we consider are based on either the First Fit (FF) or Best Fit (BF) heuristics. We use the definitions presented by Csirik and Johnson in [10]. For an algorithm that keeps at most $K$ open bins, they considered the following four rules:

1. **P-FF:** Place the current item $a$ in the lowest indexed open bin that has room for it (if any does). Otherwise open a new bin and place $a$ in it.
2. **P-BF:** Place the current item $a$ in the fullest open bin that has room for it (if any does), ties are broken in favor of the lowest index. Otherwise open a new bin and place $a$ in it.
3. **C-FF:** Close the lowest indexed open bin.

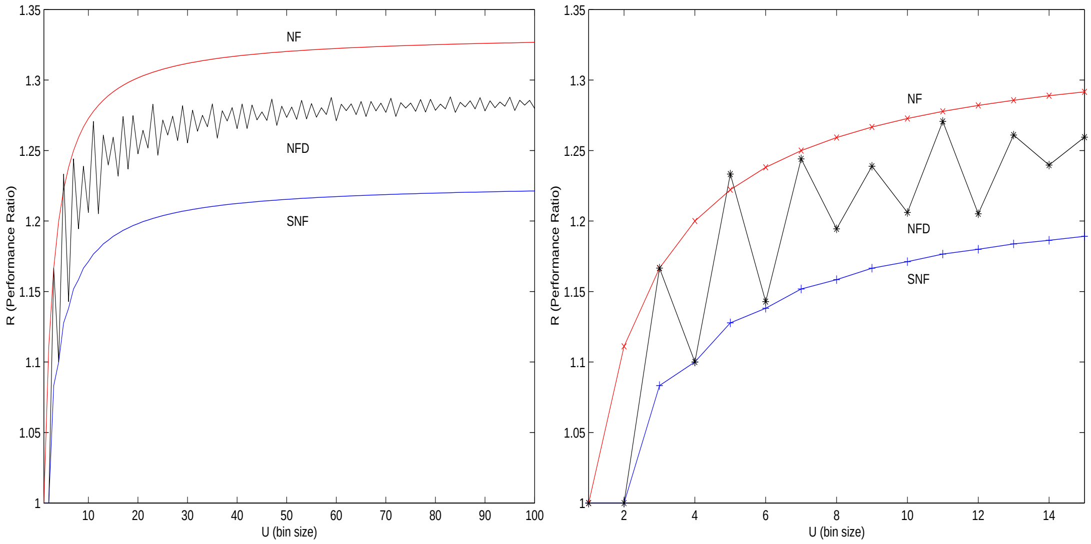



**FIG. 1.** Asymptotic expected performance ratio for distribution $\{U, U\}$.

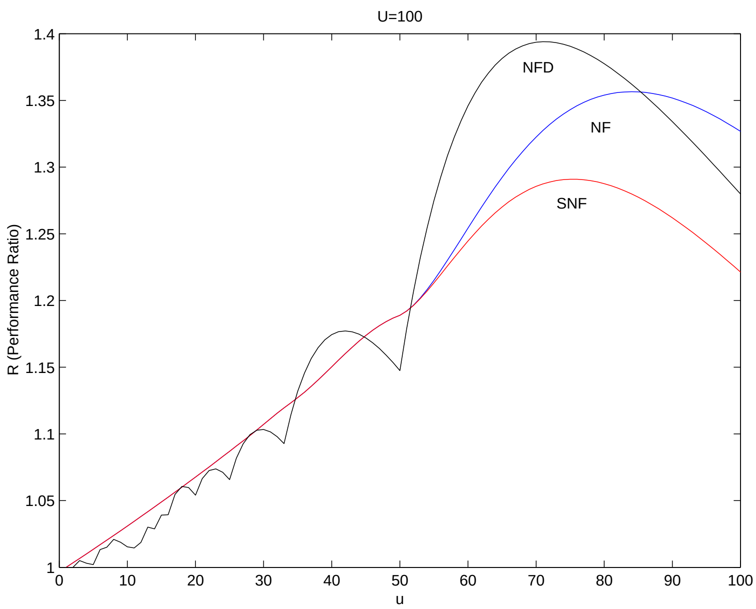



**FIG. 2.** Asymptotic expected performance ratio for distribution $\{u, U\}$ for $U = 100$.

4. **C-BF:** Close the fullest open bin, ties are broken in favor of the lowest index.

We add one more closing rule,called Smart Best Fit:

5. **C-SBF:** Close the fullest open bin among the current open bins and the bin containing $a$, ties are broken in favor of the lowest index.

Six $K$-bounded space algorithms can be constructed using any combination of a packing rule (P-FF or P-BF) with a closing rule (C-FF, C-BF or C-SBF). The algorithm packs a new item $a$, according to the packing rule. If no open bin has room for $a$ and there are $K$ open bins, the closing rule is applied and a new bin is opened. We note that the analysis of other packing and closing rules is also possible. For example, we may define the Worst Fit rules; P-WF which places the current item in the bin with the lowest level and C-WF which closes the bin with the lowest level.

The combination of P-FF with C-FF yields the *Next-K Fit* ($NF_K$) algorithm. The combination of P-BF with C-BF yields the *K-Bounded Best Fit* ($BBF_K$) algorithm and with C-SBF the *K-Smart Bounded Best Fit* ($SBBF_K$) algorithm. The other three combinations are not as interesting, we denote by $ABF_K$ the algorithm obtained by using the combination P-BF with C-FF, and by $AFB_K$ the algorithm using the combination P-FF with C-BF. The above algorithms comprise the majority of bounded space bin packing algorithms. One important class we do not consider here are the *Harmonic* algorithms $H_K$ [21], for which an average case analysis is relatively easy and has been reported in [22].

In this section we present the analysis for the $\{u, U\}$ distribution and explain how the ratio for a general distribution can be calculated. The $\{U, U\}$ distribution is a special case where $u = U$.

### 4.1. The Next-2 Fit Algorithm

Next-2 Fit ($NF_2$) uses the P-FF and C-FF rules with $K = 2$, i.e., two open bins. The next item to be packed $a$, is placed in the lowest indexed bin into which it will fit. If no open bin has room for $a$, it is placed in a new bin and (if there are already two open bins) the lowest indexed bin is closed.

The Next-$K$ Fit family of algorithms has been introduced by Johnson in [18], [19]. Csirik and Johnson presented average case results based on simulation for different values of $K$ in [10]. Our work, to the best of our knowledge, is the first analytic analysis of the algorithm.

We use the methodology developed in Section 2. To that end we denote the lowest indexed open bin by $B_1$ and the highest indexed open bin by $B_2$. The state of the packing is the content of the two open bins $N_t = (j_1,\, j_2)$, where $1 \le j_1,\, j_2 \le U$.

To calculate the transition probabilities we assume that at time $t-1$ the algorithm is in state $N_{t-1} = (j_1, j_2)$ and the next item to be packed is of size $i$. We distinguish among three cases:

1. $j_1 + i \le U$: The item fits in $B_1$. The next state is $N_t = (j_1 + i, j_2)$.

2. $j_1 + i > U$, $j_2 + i \le U$: The item does not fit in $B_1$ but fits in $B_2$. The next state is $N_t = (j_1, j_2 + i)$.

3. $j_1 + i > U$, $j_2 + i > U$: The item does not fit in $B_1$ or $B_2$. In this case $B_1$ is closed, $B_2$ becomes $B_1$ and the item is placed in a new bin which becomes $B_2$. The next state is $N_t = (j_2, i)$.

It is now possible to construct the transition matrix $P$. Once we have $P$, we can calculate the equilibrium probability vector $\Pi$.

Our next step is to calculate $OH(j_1, j_2)$, the average overhead in state $N = (j_1, j_2)$. Assume that the next item to be packed is of size $i$, $1 \le i \le U$. Note that overhead units are added only if the next item does not fit in $B_1$ or $B_2$, that is, $i > U - min\{j_1, j_2\}$. When overhead units are added the overhead is the unused space in $B_1$, which is $U - j_1$.

$$oh_i(j_1, j_2) = \begin{cases} 0 & min\{j_1, j_2\} + i \le U \\ U - j_1 & min\{j_1, j_2\} + i > U \end{cases} \tag{39}$$

For discrete uniform distribution $\{u, U\}$ the average overhead is therefore:

$$OH(j_1, j_2) = max\left\{ \frac{(u + min\{j_1, j_2\} - U) \cdot (U - j_1)}{u},\ 0 \right\} \tag{40}$$

We now use $\Pi(j_1, j_2)$ and $OH(j_1, j_2$ to calculate the asymptotic expected performance ratio in the following way:

$$\overline{R}^{\infty}_{NF_2}(\{u,U\}) = \frac{I_{av}(NF_2)}{I_{av}(OPT)} = 1 + \frac{2}{u+1}\sum_{j_1=1}^{U}\sum_{j_2=1}^{U}\Pi(j_1,j_2)\cdot OH(j_1,j_2) \tag{41}$$

To calculate the average combined size of the items for a general distribution we use the equilibrium probabilities and (39) as the expression for the overhead component.

$$I_{av}(NF_2) = \sum_{j_1=1}^{U}\sum_{j_2=1}^{U}\sum_{i=1}^{U} h_i \cdot \Pi(j_1,j_2)\cdot(i + oh_i(j_1,j_2)) \tag{42}$$

We present some computational results for the $\mathrm{NF}_2$ algorithm in Section 4.5.

### 4.2. The 2-Bounded Best Fit Algorithm

The 2-Bounded Best Fit ($BBF_2$) algorithm uses two open bins. The next item to be packed $a$, is placed in the fullest bin into which it will fit. Ties are broken in favor of the bin with lower index. If no open bin has room for $a$ and there are already two open bins, the fullest bin is closed (again, ties are broken in favor of lower index). The item is then placed in a new bin.

The $K$-Bounded Best Fit family of algorithms has been introduced and studied by Csirik and Johnson in [10]. They proved that the asymptotic worst case performance ratio of the algorithm is: $R^{\infty}_{BBF_K} = 1.7$, for any $K \geq 2$. This is interesting since it means that the worst case performance ratio of the 2-bounded space algorithm is equal to that of the unbounded Best Fit algorithm. Csirik and Johnson presented average case results based on simulation for different values of $K$ in [10]. Our work, to the best of our knowledge, is the first analytic analysis of the algorithm.

We use the content of the two open bins as the state of the packing. However, unlike $NF_2$, the indexes of the bins are of no real importance to the analysis. We can use this fact to reduce the number of states by selecting $B_1$ and $B_2$ to be the bins with the higher and lower content, respectively. The state of the packing, $N_t = (j_1, j_2)$, now has the property of $j_1 \geq j_2$, $j_1 \leq U$, which means the number of states is: $U(U+1)/2$.

To calculate the transition probabilities we assume that at time $t-1$ the algorithm is in state $N_{t-1} = (j_1, j_2)$ and the next item to be packed is of size $i$. We distinguish among three cases:

1. $j_1 + i \leq U$: The item fits in $B_1$. The next state is $N_t = (j_1 + i, j_2)$.
2. $j_1 + i > U$, $j_2 + i \leq U$: The item does not fit in $B_1$ but fits in $B_2$. If $j_1 \geq j_2 + i$ the next state is $N_t = (j_1, j_2 + i)$, otherwise the next state is $N_t = (j_2 + i, j_1)$.
3. $j_1 + i > U$, $j_2 + i > U$: The item does not fit in $B_1$ or $B_2$. In this case $B_1$ is closed and the item is placed in a new bin. If $j_2 \geq i$ the next state is $N_t = (j_2, i)$, otherwise the next state is $N_t = (i, j_2)$.

Based on the above rules, we can construct the transition matrix $P$ and calculate the equilibrium probability vector $\Pi$.

We now calculate $OH(j_1, j_2)$, the average overhead in state $N(j_1, j_2)$. Assume that the next item to be packed is of size $i$, $1 \leq i \leq U$. Note that overhead units are added only if the next item does not fit in $B_1$ or $B_2$, that is, $i > U - j_2$. When overhead units are added the overhead is the unused space in the fullest bin, that is, $U - j_1$.

$$oh_i(j_1,j_2) = \begin{cases} 0 & j_2 + i \leq U \\ U - j_1 & j_2 + i > U \end{cases} \tag{43}$$

For discrete uniform distribution $\{u, U\}$ the average overhead is therefore:

$$OH(j_1, j_2) = max\left\{\frac{(u + j_2 - U)\cdot(U - j_1)}{u},\ 0\right\} \tag{44}$$

We now use $\Pi(j_1, j_2)$ and $OH(j_1, j_2)$ to calculate the asymptotic expected performance ratio similar to (41). We use (42) to calculate the average combined size of the items for a general distribution.

We present some computational results for the $BBF_2$ algorithm in Section 4.5.

### 4.3. The Smart 2-Bounded Best Fit Algorithm

The Smart 2-Bounded Best Fit ($SBBF_2$) algorithm is similar to $BBF_2$ but includes the same improvement of Smart Next Fit over $NF$. The next item to be packed $a$, is placed in the fullest bin into which it will fit. Ties are broken in favor of the bin with lower index. If no open bin has room for $a$, it is placed in a new bin. At this point the algorithm compares the levels of the two open bins and the new bin containing $a$. The fullest bin among the three is closed (ties are broken in favor of lower index).

The Smart $K$-Bounded Best Fit algorithm is defined here for the first time. Therefore, it has not been studied before. We note that $SBBF_2$ may actually be considered as a 3-bounded space algorithm by some applications. The stage where an item is packed in a new bin which is then immediately closed, may require the space of three open bins.

Similar to $BBF_2$, we denote the bin with the higher content by $B_1$ and the bin with the lower content by $B_2$. The state of the packing, $N_t = (j_1, j_2)$, is the content of the open bins.

To calculate the transition probabilities we assume that at time $t-1$ the algorithm is in state $N_{t-1} = N(j_1, j_2)$ and the next item to be packed is of size $i$. We distinguish among three cases:

1. $j_1 + i \leq U$: The item fits in $B_1$. The next state is $N_t = (j_1 + i, j_2)$.
2. $j_1 + i > U$, $j_2 + i \leq U$: The item does not fit in $B_1$ but fits in $B_2$. If $j_1 \geq j_2 + i$ the next state is $N_t = (j_1, j_2 + i)$, otherwise the next state is $N_t = (j_2 + i, j_1)$.
3. $j_1 + i > U$, $j_2 + i > U$: The item does not fit in $B_1$ or $B_2$. In this case $B_1$ is closed and the item is placed in a new bin. If $i \geq j_1$ the new bin is closed and the next state remains $N_t = (j_1, j_2)$, otherwise $B_1$ is closed and the next state is $N_t = (j_1', j_2')$ where $j_1' = max\{j_2, i\}$ and $j_2' = min\{j_2, i\}$.

We can now construct the transition matrix $P$ and calculate the equilibrium probabilities.

Our next step is to calculate $OH(j_1, j_2)$. Assume that the next item to be packed is of size $i$, $1 \leq i \leq U$. Note that overhead units are added only if the next item does not fit in $B_1$ or $B_2$, that is, $i > U - j_2$. When overhead units are added the overhead is the unused space in the fullest bin, that is $U - max\{j_1, j_2, i\}$.

$$oh_i(j_1, j_2) = \begin{cases} 0 & j_2 + i \leq U \\ U - max\{j_1, j_2, i\} & j_2 + i > U \end{cases} \tag{45}$$

For discrete uniform distribution $\{u, U\}$ the average overhead is therefore:

$$OH(j_1, j_2) = \begin{cases} 0 & j_2 + u \leq U \\ \frac{1}{u}\sum_{i=U-j_2+1}^{u} U - max\{j_1, j_2, i\} & j_2 + u > U \end{cases} \tag{46}$$

We now use $\Pi(j_1, j_2)$ and $OH(j_1, j_2)$ to calculate the asymptotic expected performance ratio similar to (41). We use (42) to calculate the average combined size for a general distribution.

We present some computational results for the $SBBF_2$ algorithm in Section 4.5.

### 4.4. The ABF2 and AFB2 Algorithms

The $ABF_K$ and $AFB_K$ algorithms are hybrids of $NF_K$ and $BBF_K$. $ABF_K$ algorithms use the P-BF packing rule, i.e., an item is placed in the fullest bin, and the C-FF closing rule, i.e., if the item does not fit in any open bin, the bin with the lowest index is closed. $AFB_K$ algorithms use the P-FF packing rule and the C-BF closing rule. We do not present the analysis of the algorithms in details, since their analysis is very similar to that of $NF_2$ and $BBF_2$.

$ABF_K$ algorithms have one important advantage over $BBF_K$; they guarantee "bounded delay", that is, the bin into which an item is packed is closed after at most $K-1$ other bins have been closed. Such bounded delay may be required (or just desirable) by several applications. As we can see from Table 2, the performance ratio of the $ABF_2$ is only slightly better than that of $NF_2$. We note that the worst case performance ratio of $ABF_K$ is also slightly better than that of $NF_K$ for any $K \geq 2$ [23]. $AFB_K$ algorithms do not have a bounded delay but perform better than $ABF_K$ on average (their worst case ratio is similar to that of $NF_K$ [28]). The expected performance ratio of the $AFB_2$ algorithm lies somewhere between the performance ratio of $NF_2$ and $BBF_2$ (see Table 2).

### 4.5. Summary of Results

In this section we present some computational results of the expected performance ratio of 2-bounded space algorithms. In Table 2 we present results for distribution $\{U, U\}$ for several values of $U$. The results for $U \leq 300$ were computed using the analysis we presented in the previous subsections. The value for $U \to \infty$ is estimated since we could not obtain numeric results for very large values of $U$. To get an estimation we studied the behavior of the performance ratio for values of $U \leq 300$ and estimated its asymptotic value. Our estimation is enhanced by the fact that the $NF$ and $SNF$ algorithms have a similar asymptotic behavior, with the same difference (0.002) between the ratio for $U = 300$ and $U \to \infty$. It is therefore reasonable to believe that the estimation error is less than 0.001. Our estimation of the expected performance ratio for $U \to \infty$ agree with the simulation result, for the continuous uniform distribution [0,1], reported in [10].

**TABLE 2.**

Asymptotic expected performance ratio for distribution $\{U, U\}$.

| U | $NF_2$ | $BBF_2$ | $SBBF_2$ | $ABF_2$ | $AFB_2$ |
|---|---|---|---|---|---|
| 5 | 1.1407 | 1.0939 | 1.0801 | 1.1402 | 1.1012 |
| 10 | 1.1836 | 1.1264 | 1.1093 | 1.1811 | 1.1391 |
| 20 | 1.2095 | 1.1508 | 1.1329 | 1.2070 | 1.1641 |
| 50 | 1.2265 | 1.1665 | 1.1485 | 1.2241 | 1.1805 |
| 100 | 1.2326 | 1.1725 | 1.1542 | 1.2301 | 1.1867 |
| 300 | 1.2367 | 1.1763 | 1.1580 | 1.2341 | 1.1906 |
| $\infty$ | 1.2386 | 1.1783 | 1.1600 | 1.2362 | 1.1926 |

The graphs in Figure 3 present the expected performance ratio for the $\{U, U\}$ distributions for all values of $U \leq 50$. As a rule we can say that the performance ratio of all algorithms is monotonic increasing with $U$. Note however that $BBF_2$ and $SBBF_2$ have an exception to this rule; the value for $U = 3$ is actually higher than that of $U = 4$. We can see that Best-Fit performs better than Next-Fit for any value of $U$. The Smart Bounded Best-Fit algorithm achieves the best results among all 2-bounded space algorithms. Its performance ratio is lower than that of $BBF$ for all values of $U$ and the difference is almost constant (about 0.017). The lower ratio is due to the fact that $SBBF$ packs large items more efficiently. Recall however that $SBBF_2$ may actually be considered as a 3-bounded space algorithm by some applications.

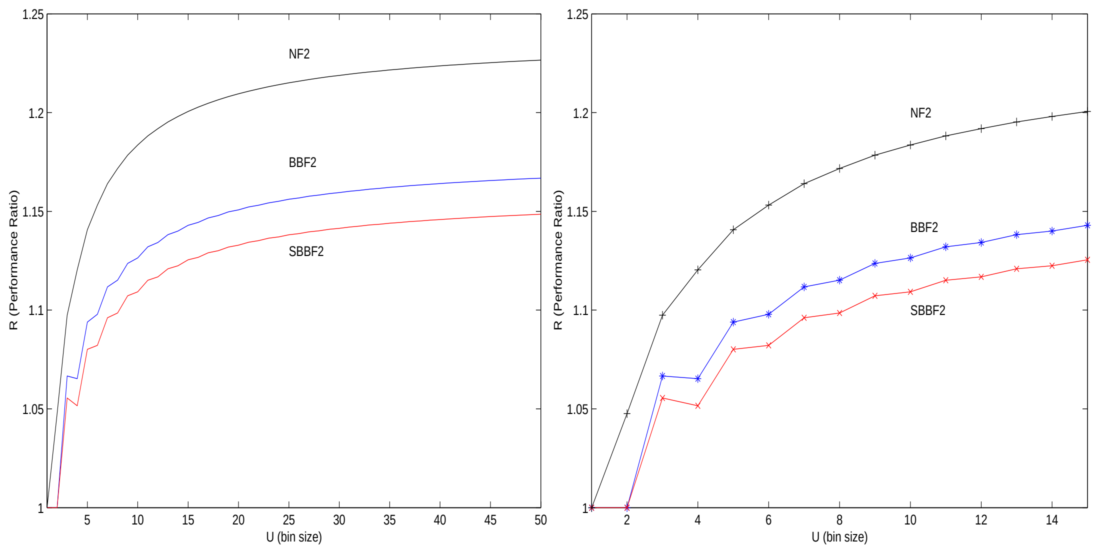



**FIG. 3.** Asymptotic expected performance ratio for distribution $\{U, U\}$.

Figure 4 presents the expected performance ratio for the $\{u, U\}$ distribution, when $U = 50$, for values of $u \leq 50$. Other values of $U$ produce similar graphs, that is, the shape of the curves remains the same. We observe that Best-Fit is better than First-Fit for all values of $u$. Combining this observation with the results for the $\{U, U\}$ distribution (Figure 3) we conclude that Best-Fit is superior for any $\{u, U\}$ distribution. As we expect, the performance ratio of $BBF_2$ and $SBBF_2$ is identical for all values of $u \leq U/2$. In Figure 5 we compare the expected performance ratio of 2-bounded space algorithm with that of 1-bounded space algorithms. The pattern of the curve of the $NF_2$ algorithm is similar to that of $NF$ but with considerably lower values of performance ratio. The performance ratio of all algorithms has a maximum point around $u = 0.85U$.

## 5. ANALYSIS OF K-BOUNDED SPACE ALGORITHMS WITH $K > 2$

In the previous section we presented a detailed analysis of the algorithms $NF_2$, $BBF_2$ and $SBBF_2$. We also considered the $ABF_2$ and $AFB_2$ algorithms. The analysis of the same algorithms, for higher values of $K$ is very similar. The state of the packing is the content of the $K$ open bins $N_t = (j_1, j_2, ..., j_K)$, where $1 \leq j_1, j_2, ..., j_K \leq U$. To calculate the expected performance ratio, we first use the packing and closing rules of the algorithm to construct the transition matrix $P$. Once we have the transition matrix we can calculate the equilibrium probabilities $\Pi(j_1, j_2, ..., j_K)$. Next we calculate the overhead component $oh_i(j_1, j_2, ..., j_K)$. Finally we calculate the average combined size of the items in the following way:

$$I_{av}(A_K) = \sum_{j_1=1}^{U} \cdots \sum_{j_K=1}^{U} \sum_{i=1}^{U} h_i \cdot \Pi(j_1, ..., j_K) \cdot (i + oh_i(j_1, ..., j_K)) \tag{47}$$

To find the expected performance ratio we divide (47) by the combined average size of the optimal packing, which is $I_{av}(OPT) = \frac{u+1}{2}$ for the uniform distribution $\{u, U\}$.

The main drawback of the analysis is that the number of possible states of the algorithm increases as $O\left(U^K\right)$. Since we rely on numerical computation of the equilibrium probabilities,

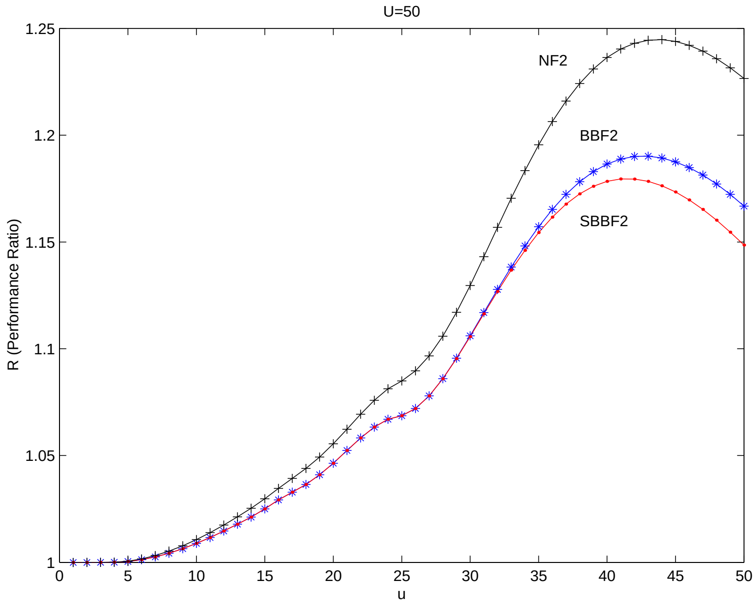



**FIG. 4.** Asymptotic expected performance ratio for distribution $\{u, U\}$ for $U = 50$.

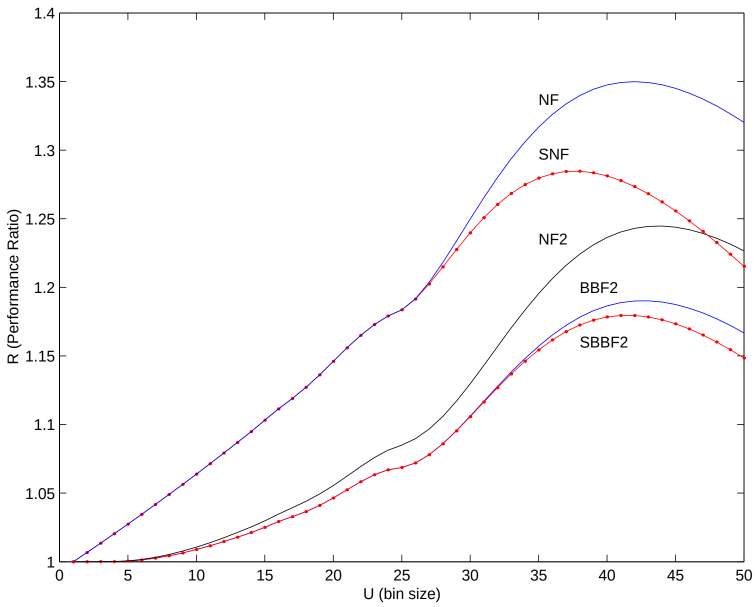
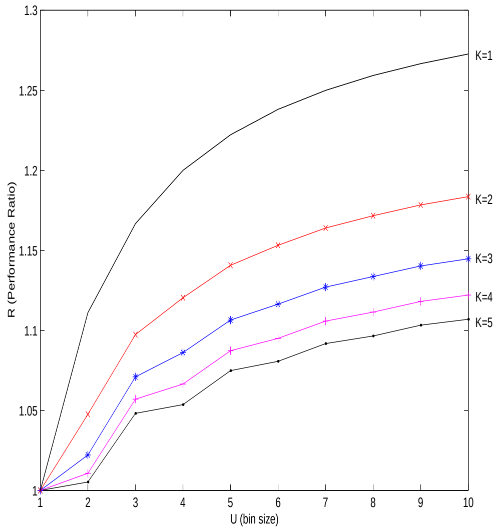



**FIG. 5.** Comparison of some bounded space algorithms for distribution $\{u, U\}$ for $U = 50$.

this means that calculating the performance ratio for higher values of $K$ can only be done for very small values of $U$.

Figure 5 shows the expected performance ratio of the Next-$K$ Fit and $K$-Bounded Best-Fit algorithms for different values of $K$, distribution $\{U, U\}$ and values of $U \leq 10$. As we expect the performance ratio is decreasing with $K$. Note however that the improvement obtained by adding an additional bin is decreasing with $K$; the difference between $K = 5$ and $K = 4$ is not as significant as the difference between $K = 2$ and $K = 1$. It is interesting to note that as $K$

increases the performance ratio of $BBF_k$ becomes less monotonic increasing with $U$. For small bin sizes the algorithm performs better when $U$ is even compared to the odd value of $U-1$.



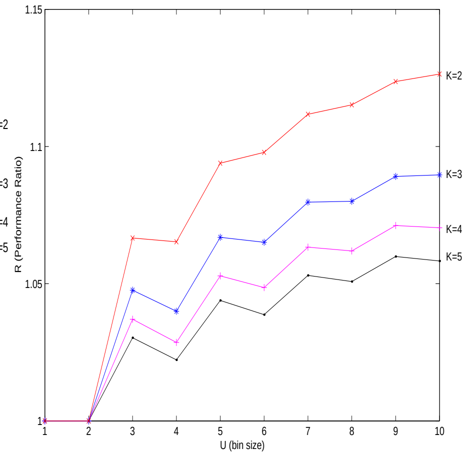




**FIG. 6a.** Next-$K$ Fit Performance ratio for values of $K \leq 5$, distribution $\{U, U\}$.

**FIG. 6b.** $K$-Bounded Best Fit Performance ratio for values of $2 \leq K \leq 5$, distribution $\{U, U\}$.

## 6. CONCLUDING REMARKS

In this paper we presented an average case analysis of most well known bounded space bin packing algorithms. The analysis is based on a new technique of average case analysis in which the expected performance ratio of an algorithm is derived from the average combined size of the items. The packing of the algorithm is modeled by a Markov chain and the combined size of an item is calculated from its actual size plus the overhead (wasted space) it creates.

Our technique of average case analysis has several advantages: it is suitable for analyzing any (i.i.d) item size distribution, it can be applied to a wide variety of algorithms and the calculation is relatively simple. The main drawback of the analysis lies in its computational complexity. The number of possible states of the Markov chain increases as $O\left(U^K\right)$, which renders the analysis of large values of $K$ and $U$ impractical. It seems that there is no way around this complexity problem if an exact numerical computation is needed. Providing a way of calculating or, more likely, approximating the expected performance ratio for higher values of $K$ is a subject we leave for future research.

## APPENDIX A

### Equilibrium Probabilities

Our average case analysis is based on modeling the packing of an algorithm by a Markov chain. To apply the analysis we assume the Markov chain has equilibrium probabilities with ergodic properties. In this section we justify this assumption by showing that, for most distributions, the Markov chain describing any of the algorithms has unique equilibrium probabilities. To that end, we study the class structure of such Markov chains.

Recall that for a $K$-bounded space algorithm, the state of the packing, $N_t(j_1, j_2, ..., j_K)$, is the content of the open bins after $t$ items were packed. Since the bin size is $U$, there can be at most $U^K$ states. The chain is therefore finite for every finite $U$ and $K$. In a finite Markov chain every irreducible aperiodic class is ergodic and has unique equilibrium probabilities [24]. The equilibrium probabilities are independent of the initial state.

Let $A_K$ denote any online algorithm considered in this paper, that is, $A_K$ is one of the algorithms $NF_K$, $SNF$, $BBF_K$, $SBBF_K$, $ABF_K$ or $AFB_K$. The following theorem establishes sufficient conditions for the existence of unique equilibrium probabilities.

THEOREM A.1. *Let $H$ be any distribution containing items of sizes one and two, i.e., $h_1, h_2 > 0$. The Markov chain describing the packing of algorithm $A_K$ for distribution $H$ has unique equilibrium probabilities.*

*Proof.* We begin by identifying one ergodic state. Denote by $S_U$ the state where all the open bins are full, i.e., $N_t(j_1, j_2, ...j_K) = (U, U, ..U)$. We now show that $S_U$ is ergodic.

*Claim A.1.* State $S_U$ is ergodic.

*Proof.* We show that $S_U$ is positive recurrent and aperiodic, hence ergodic. Note that any state $s$ leads to $S_U$ by a series of items of size one that fill all bins. State $S_U$ is therefore recurrent, it is positive recurrent since in a finite Markov chain all recurrent states are positive recurrent. We now show that $S_U$ is aperiodic. The probability to go from $S_U$ to $S_U$ in $n$ steps, $P^{(n)}_{S_U,S_U}$, is positive for any $n \geq \lceil U/2 \rceil$. The transition can be done by selecting items of sizes one and two only. For example, to go from $S_U$ to $S_U$ in $n = U$ steps choose $U$ items of size one, to go in $n = U - 1$ steps choose one item of size two and $U - 2$ items of size one. State $S_U$ is thus aperiodic since $P^{(n)}_{S_U,S_U} > 0$ for all sufficiently large $n$ [24]. ▮

So far we proved the existence of one ergodic state, $S_U$. We know that if state $s$ communicates with $S_U$, i.e., $s \leftrightarrow S_U$, $s$ is also an ergodic state. Note that since every state leads to $S_U$, there can be only one subset of ergodic states in the chain. A state $s$ is therefore ergodic if and only if it is accessible from $S_U$; all other states are transient. The subset of states that communicate with $S_U$ form a non empty ergodic class which is the only irreducible class of the Markov chain. This property ensures the existence of unique equilibrium probabilities. ▮

COROLLARY A.1. *For the $\{u, U\}$ distribution the Markov chain of $A_K$ has unique equilibrium probabilities, for any $u \geq 2$.*

*Proof.* Immediate from Theorem A.1. ▮

*Remark about periodic chains:* Note that for the distribution $\{1, U\}$ the chain is not ergodic since it is periodic (with period $U$). However, the chain has only one irreducible class and therefore has unique equilibrium probabilities. In general, any finite Markov chain with a single irreducible class (even if it is periodic) has unique equilibrium probabilities and our analysis can be applied to it.

It is important to note that Theorem A.1 provides sufficient conditions only. In many cases the equilibrium probabilities exists even if the conditions set by Theorem A.1 do not hold. In fact, it takes some effort to find an example where the Markov chain does not have unique equilibrium probabilities. The following is such an example.

***Example:*** Let $U = 5$ and let $h_4 = h_5 = 0.5$. Consider the $NF_2$ and $BBF_2$ algorithms. It is easy to verify that the chain describing $NF_2$ contains only one irreducible class: {(4,4), (4,5), (5,4), (5,5)}. On the other hand, the chain describing $BBF_2$ contains three irreducible classes: {(4,4), (5,4), (5,5)}, {(4,2), (5,2)} and {(4,3), (5,3)}. Note that the initial state (all bins are empty) leads only to the first class. In order to perform the analysis in this case it is necessary to apply it only to the irreducible class to which the initial state leads.

In most cases there is no need to identify the ergodic states in order to perform the analysis. The exceptions are cases, such as the one presented in the above example, where the chain contains more than one irreducible class. However, identifying the ergodic states enables us to reduce the number of states considered in the analysis, resulting in a more efficient numeric calculation of the equilibrium probabilities. In the next subsection we identify the ergodic states for the $\{u, U\}$ distribution.

## A.1. ERGODIC STATES OF THE UNIFORM DISTRIBUTION

It is easy to verify that in 1-bounded space algorithms, i.e., $NF$ and $SNF$, all states communicate and are therefore ergodic. For $K \geq 2$ not all states are ergodic. The subset of ergodic states depends on the algorithm.

### A.1.1. $NF_K$ Algorithm

Let us first consider the $NF_2$ algorithm. A state $N(j_1, j_2)$ is ergodic if:

1. $j_1 + min\{j_2, u\} > U$
2. $j_2$ is a combination of items with sizes from the set $\{U - j_1 + 1, U - j_1 + 2, ..., u\}$.

All other states are transient and can only be visited during the first stage of the algorithm, before the first two bins are closed. For example, let $U = 10$, $u = 5$ and consider two states $N'(j_1, j_2) = (8, 4)$ and $N''(j_1, j_2) = (5, 4)$. Assuming we start with two empty bins we can reach both states. However, there is a positive probability to return to state $N'$ (for example if the next two items are of size four) while the probability to return to $N''$ is zero. Therefore $N'$ is ergodic while $N''$ is transient.

For $K > 2$ a state $N(j_1, j_2, ...j_K)$ is ergodic if:

1. For every $1 \leq x < K$, $j_x + min\{j_{x+1}, j_{x+2}, ..., j_K, u\} > U$
2. For every $2 \leq x < K$, $j_x$ is a combination of items with sizes from the set $\{b, b+1, ..., u\}$, where $b = U + 1 - min\{j_i : \ 1 \leq i < x\}$.

### A.1.2. $BBF_K$ and $SBBF_K$ Algorithms

Recall that we defined the state of the packing $N(j_1, j_2, ..., j_K)$, such that $j_1 \geq j_2 \geq ... \geq j_K$. When using this order, the ergodic states of the algorithms are very similar to those of $NF_K$. A state $N(j_1, j_2, ..., j_K)$ is ergodic if:

1. $j_1 \geq j_2 \geq ... \geq j_K$.
2. For every $1 \leq x < K$, $j_x + min\{j_{x+1}, j_{x+2}, ..., j_K, u\} > U$
3. For every $2 \leq x < K$, $j_x$ is a combination of items with sizes from the set $\{b, b+1, ..., u\}$, where $b = U + 1 - j_{x-1}$.

## APPENDIX B

### Analysis of NF for the $\{u, U\}$ Distribution

In Section 3.1.2 we analyzed the $NF$ algorithm for the $\{u, U\}$ distribution. Unlike the case of the $\{U, U\}$ distribution we did not give a closed form solution because we did not have a closed form solution for the equilibrium probabilities. In this section we elaborate more on the subject and show how to find the equilibrium probabilities or at least get a good approximation.

The transition matrix of the Markov chain define the following equilibrium equations:

$$\Pi_1 = \frac{1}{u}\Pi_U \tag{B.1}$$

$$\Pi_k = \frac{1}{u}\sum_{j=1}^{k-1}\Pi_j + \frac{1}{u}\sum_{j=U-k+1}^{U}\Pi_j \qquad 2 \le k \le u \tag{B.2}$$

$$\Pi_k = \frac{1}{u}\sum_{j=k-u}^{u-1}\Pi_j \qquad u+1 \le k \le U \tag{B.3}$$

We are interested in the equilibrium probabilities form which we can calculate the asymptotic expected performance ratio:

$$\overline{R}_{NF}^{\infty}(\{u, U\}) = 1 + \frac{2}{u+1}\sum_{j=U-u+1}^{U}\Pi_j\frac{(j+u-U)(U-j)}{u} \tag{B.4}$$

The method of the calculation depends on the value of $u$ (maximum item size). We therefore distinguish between cases where $u > \frac{U}{2}$ and cases where $u < \frac{U}{2}$.

### Calculations for the Range $\frac{U}{2} > u \le U$

For $u > \frac{U}{2}$ there is a simple expression for the probabilities $\Pi_{U-u}, \dots, \Pi_u$:

$$\Pi_j = \frac{2j-(U-u)}{u(u-1)} \qquad U-u \le j \le u \tag{B.5}$$

Using (B.5) we can calculate part of the sum in (B.4):

$$\begin{aligned}\sum_{j=U-u+1}^{u}\Pi_j\, OH(j) &= \sum_{j=U-u+1}^{u}\frac{2j-(U-u)}{u(u-1)}\,\frac{(j+u-U)(U-j)}{u} \\ &= \frac{(2u-U)(2u-U+1)\left((2U-3u)(1+u-U)+U^2\right)}{6u^2(u+1)}\end{aligned} \tag{B.6}$$

Using (B.6) and (B.6) it is relatively easy to find the expected performance ratio when $u$ is close to $U$.

*Calculating the Expected Performance Ratio for Distributions $\{U-1, U\}$ and $\{U-2, U\}$*

In the case of $u = U - 1$ we know, form (B.5), all the equilibrium probabilities that we need and the calculation is straightforward:

$$I_{av}(NF) = \frac{u+1}{2} + \sum_{j=2}^{U-1}\Pi_j \cdot OH(j) = \frac{U}{2} + \frac{(U-2)U}{6(U-1)} \tag{B.7}$$

$$\overline{R}_{NF}^{\infty}(\{U-1,U\}) = \frac{I_{av}(NF)}{U/2} = 1 + \frac{U-2}{3(U-1)} = \frac{4U-5}{3(U-1)} \tag{B.8}$$

Note that, as we expect, when $U \to \infty$ we get the same ratio as for the $\{U,U\}$ distribution: $\overline{R}_{NF}^{\infty}(\{U-1,U\}) = \frac{4}{3}$.

We now turn to the $\{U-2,U\}$ distribution. Here we know the probabilities $\Pi_2 \ldots \Pi_{U-2}$ but we must find $\Pi_{U-1}$ in order to compute the expected performance ratio. We find $\Pi_{U-1}$ using (B.1)-(B.3), since we know the value of most of the probabilities the calculation is easy. We find that $\Pi_{U-1} = \frac{U^2-4U+5}{(U^2-3U+3)(U-1)}$. We now substitute $u = U-2$ in (B.6) and obtain the following expression for $I_{av}(NF)$:

$$I_{av}(NF) = \frac{U-1}{2} + \frac{(U-4)(U^2-9)}{6(U-2)(U-1)} + \frac{(U^2-4U+5)(U-3)}{(U^2-3U+3)(U-1)(U-2)} \tag{B.9}$$

We can now calculate the expected performance ratio:

$$\begin{aligned} \overline{R}_{NF}^{\infty}(\{U-2,U\}) &= \frac{I_{av}(NF)}{(U-1)/2} \\ &= 1 + \frac{(U-4)(U^2-9)}{3(U-2)(U-1)^2} + \frac{(U^2-4U+5)(U-3)}{(U^2-3U+3)(U-1)^2(U-2)} \end{aligned} \tag{B.10}$$

*Approximation for the range* $\frac{U}{2} < u \leq U$

As we could see, the calculation of the expected performance ratio is getting harder as $u$ decreases. In this section we provide an approximation of the expected performance ratio for the range $\frac{U}{2} < u \leq U$. From (B.5) we know the exact value of the probabilities $\Pi_{U-u}, \ldots, \Pi_u$. Based on our numerical results we approximate the probabilities $\Pi_{u+1}, \ldots, \Pi_U$ to be:

$$\Pi_j = \frac{1}{2}\Pi_u = \frac{3u-U}{2u(u-1)}, \quad u+1 \leq j \leq U \tag{B.11}$$

Using the above approximation we can calculate the following sum:

$$\begin{aligned} \sum_{j=u+1}^{U} \Pi_j OH(j) &= \sum_{j=u+1}^{U} \frac{3u-U}{2u(u+1)} \frac{(j+u-U)}{u} \\ &= \frac{3u-U}{2u^2(u+1)} \sum_{j=1}^{U-u} (j+2u-U)(U-j-u) = \frac{(3u-U)(u-1)}{2u} \end{aligned} \tag{B.12}$$

We obtain the approximation for the expected performance ratio by adding (B.6) and (B.12).

$$\begin{aligned} \overline{R}_{NF}^{\infty}(\{u,U\}) \approx 1 &+ \frac{(2u-U)(2u-U+1)\left((2U-3u)(1+u-U)+U^2\right)}{6u^2(u+1)} \\ &+ \frac{(3u-U)(u-1)}{2u} \end{aligned} \tag{B.13}$$

If $U$ is sufficiently large ($U > 10$), the approximation is very accurate when $u$ is close to $U$. The approximation is less accurate when $u$ is close to $\frac{U}{2}$.

**Calculations for the Range $u \ll U$**

We start by analyzing the distribution $\{2, U\}$, i.e., $u = 2$. Equations (B.1)-(B.3) now become:

$$\Pi_1 = \frac{1}{2}\Pi_U \tag{B.14}$$

$$\Pi_2 = \frac{1}{2}(\Pi_{U-1} + \frac{3}{2}\Pi_U) \tag{B.15}$$

$$\Pi_k = \frac{1}{2}(\Pi_{k-1} + \Pi_{k-2}), \quad 3 \le k \le U \tag{B.16}$$

We use a generating function to find the probabilities:

$$\begin{aligned}\Pi(z) &= \sum_{k=1}^{U} \Pi_k z^k = \Pi_1 z + \Pi_2 z^2 + \frac{1}{2}\sum_{k=3}^{U}(\Pi_{k-1} + \Pi_{k-2})z^k \\ &= \Pi_1 z + \Pi_2 z^2 + \frac{1}{2}z(\Pi(z) - \Pi_1 z - \Pi_U z^U) + \frac{1}{2}z^2(\Pi(z) - \Pi_{U-1}z^{U-1} - \Pi_U z^U)\end{aligned} \tag{B.17}$$

$$\Pi(z) = \frac{(z^{U+2} + z^{U+1} - z^2 - z)\Pi_U + (z^{U+1} - z^2)\Pi_{U-1}}{z^2 + z - 2} \tag{B.18}$$

We now have $\Pi(z)$ as a function of $\Pi_U$ and $\Pi_{U-1}$. To find the probabilities we use two properties of the generating function:

1. $\Pi(z = 1) = 1$. This property gives us the equation:

$$\Pi(z = 1) = \frac{(U + 2 + U + 1 - 2 - 1)\Pi_U + (U + 1 - 2)\Pi_{U-1}}{2 + 1} = 1$$

from which we obtain: $\Pi_{U-1} = \frac{3 - 2U\Pi_U}{U-1}$.

2. Since the generating function is analytic for any value of $z$, any root of the denominator must also be a root of the numerator. It is easy to verify that $z = -2$ is such a root, we therefore get another equation:

$$\left((-2)^{U+2} + (-2)^{U+1} - (-2)^2 - (-2)\right)\Pi_U + \left((-2)^{U+1} - (-2)^2\right)\Pi_{U-1} = 0$$

Using the above equations we obtain:

$$\Pi_U = \left[\frac{(U-1)\left((-2)^{U+1} - 1\right)}{6\left((-2)^{U-1} - 1\right)} + \frac{U+1}{3}\right]^{-1} \tag{B.19}$$

$$\Pi_{U-1} = \frac{3 - 2U\Pi_U}{U - 1}$$

Once we have $\Pi_{U-1}$ we can calculate the expected performance ratio:

$$\overline{R}^{\infty}_{NF}(\{2, U\}) = 1 + \frac{2}{3}\Pi_{U-1} \cdot OH(U - 1) = 1 + \frac{1}{3}\Pi_{U-1} \tag{B.20}$$

We observe that when $U$ is sufficiently large it is possible to get a very good approximation by using: $\Pi_U \cong \Pi_{U-1} \cong \frac{3}{3U-1}$. We therefore get the following approximation of the expected performance ratio:

$$\overline{R}^{\infty}_{NF}(\{u, U\}) \approx \frac{3U}{3U - 1} \tag{B.21}$$

*Approximation for the range* $u < \frac{U}{2}$

We now generalize the analysis we performed for $u = 2$ to other values of $u < \frac{U}{2}$. We can derive a generating function for any value of $u$, in a similar way as we did for $u = 2$. The number of unknown probabilities in the generating function is $u$ and the generating function has the following format:

$$\Pi(z) = \frac{(\sum_{k=2}^{u} z^k - zu)\Pi_1 + (\sum_{k=3}^{u} z^k - z^2 u)\Pi_2 + ... + (\sum_{k=u}^{u} z^k - z^{u-1}u)\Pi_{u-1}}{\sum_{k=1}^{u} z^k - u} + \frac{-z^u u\Pi_u + \sum_{k=U+1}^{U+u} z^k \Pi_U + \sum_{k=U+1}^{U+u-1} z^k \Pi_{U-1} + ... + z^{U+1}\Pi_{U+1-u}}{\sum_{k=1}^{u} z^k - u} \tag{B.22}$$

An exact calculation of the probabilities gets harder as $u$ increases but, similar to the case of $u = 2$, it is possible to get a good approximation if we assume that $\Pi_U \cong \Pi_{U-1} \cong ... \cong \Pi_u$. In this case we have:

$$\Pi_U \cong \Pi_{U-1} \cong ... \cong \Pi_u \cong \frac{3u(u+1)}{3u(u+1)U - u(u+1)(u-1)} \tag{B.23}$$

Using (B.23) we obtain the following approximation of the expected performance ratio:

$$\begin{aligned} \overline{R}_{NF}^{\infty}(\{u, U\}) &\approx 1 + \frac{2}{u+1} \sum_{j=U-u+1}^{U} \frac{(j-u+U)(U-j)}{u} \frac{3}{3U-u+1} \\ &= 1 + \frac{2}{u+1} \sum_{j=1}^{u} \frac{j(u-j)}{u} \frac{3}{3U-u+1} = \frac{3U}{3U-u+1} \end{aligned} \tag{B.24}$$

If we combine the approximation we derived in (B.13) for $u > \frac{U}{2}$, with the approximation given in (B.24) for $u < \frac{U}{2}$, we get a good approximation for all values of $u$. If $U$ is sufficiently large, the approximation error is negligible when $u \approx 2$ or $u \approx U$. As $u$ becomes closer to $\frac{U}{2}$ the approximation error increases. In Figure 1 we present the approximation we obtain by using (B.24) and (B.13) for distribution $\{u, U\}$ when $U = 50$. The approximation error in this case is no more then 0.01.

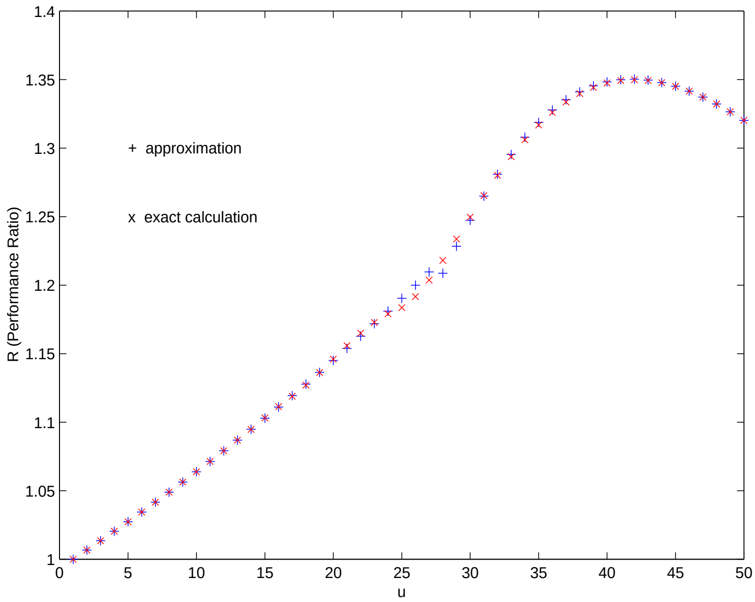



**FIG. 1.** Approximation of $NF$ vs. the exact calculation for distribution $\{u, U\}$ for $U = 50$.

## APPENDIX C

### Calculating the Expected Performance Ratio using Ratio of Expectations

In Section 2 we use a property of the optimal packing that ensures that, for any item size distribution, as $n \to \infty$, $E[A(L_n)/OPT(L_n)]$ and $E[A(L_n)]/E[OPT(L_n)]$ converge to the same limit. In this section we prove this property.

The expected performance ratio of algorithm $A$ with item distribution $H$ is defined as follows:

$$\overline{R}_A^n(H) \equiv E\left[R_A(L_n)\right] = E\left[\frac{A(L_n)}{OPT(L_n)}\right] \tag{C.1}$$

The asymptotic expected performance ratio is defined as:

$$\overline{R}_A^\infty(H) \equiv \lim_{n\to\infty} \overline{R}_A^n(H) \tag{C.2}$$

We want to show that the ratio of expectations converges to the same limit as $n \to \infty$. We know the following properties hold for $A(L_n)$ and $OPT(L_n)$:

- $OPT(L_n) \le A(L_n) \le n$.
- $A(L_n) \le C \cdot OPT(L_n)$ ($C \le 2$ for any practical algorithm).
- The optimal packing has the following property under any distribution (see [27]):

$$Pr\left(|\,OPT(L_n) - E[OPT(L_n)]\,| \ge t\right) \le 2e^{\frac{-t^2}{2n}} \tag{C.3}$$

We are going to prove the following claim.

*Claim C.1.* For any distribution $H$ and any reasonable algorithm $A$:

$$\lim_{n\to\infty}\left|E\left[\frac{A(L_n)}{OPT(L_n)}\right] - \frac{E[A(L_n)]}{E[OPT(L_n)]}\right| = 0 \tag{C.4}$$

Proof:

$$\lim_{n\to\infty}\left| E\left[\frac{A(L_n)}{OPT(L_n)}\right] - \frac{E[A(L_n)]}{E[OPT(L_n)]}\right|$$

$$= \lim_{n\to\infty}\left| E\left[\frac{A(L_n)}{OPT(L_n)}\right] - E\left[\frac{A(L_n)}{E[OPT(L_n)]}\right]\right|$$

$$= \lim_{n\to\infty}\left| E\left[\frac{A(L_n)}{OPT(L_n)}\left[1 - \frac{OPT(L_n)}{E[OPT(L_n)]}\right]\right]\right|$$

$$\leq \lim_{n\to\infty} E\left[\left|\frac{A(L_n)}{OPT(L_n)}\left[1 - \frac{OPT(L_n)}{E[OPT(L_n)]}\right]\right|\right]$$

$$= \lim_{n\to\infty} E\left[\left|\frac{A(L_n)}{OPT(L_n)}\right| \cdot \left|\frac{E[OPT(L_n)] - OPT(L_n)}{E[OPT(L_n)]}\right|\right]$$

$$\leq \lim_{n\to\infty} R_A^n \cdot \frac{1}{n\mu} \cdot E\left[\mid E[OPT(L_n)] - OPT(L_n)\mid\right]$$

Where $R_A^n$ is the worst case performance ratio of algorithm $A$, for lists of length $n$ and $\mu$ is the mean size of the items of distribution $H$.

We now use the property that if $Y$ is a positive random variable then:

$$E[Y] = \int_0^\infty Pr(Y \geq t)dt$$

We use the above property to continue the evaluation.

$$\lim_{n\to\infty} R_A^n \cdot \frac{1}{n\mu} \cdot E\left[\mid E[OPT(L_n)] - OPT(L_n)\mid\right]$$

$$\leq \lim_{n\to\infty} R_A^n \cdot \frac{1}{n\mu}\int_0^\infty Pr\left(\mid E[OPT(L_n)] - OPT(L_n)\mid \geq t\right) dt$$

$$\leq \lim_{n\to\infty} R_A^n \cdot \frac{1}{n\mu}\int_0^\infty 2e^{\frac{-t^2}{2n}} dt = \lim_{n\to\infty} R_A^n \cdot \frac{1}{n\mu}\sqrt{2n\pi}$$

**Conclusion**: The claim holds for any distribution $H$ and any algorithm $A$ with a worst case ratio $R_A^n = o(\sqrt{n})$. In such cases we have:

$$\lim_{n\to\infty} E\left[\frac{A(L_n)}{OPT(L_n)}\right] = \lim_{n\to\infty}\frac{E[A(L_n)]}{E[OPT(L_n)]} \tag{C.5}$$

∎

## REFERENCES

1. E. G. Coffman Jr., M. R. Garey, and D. S. Johnson. Approximation algorithms for bin-packing: An updated survey. In G. Ausiello, M. Lucertini, and P. Serafini, editors, Algorithm Design for Computer System Design, pages 49-106. Springer-Verlag, Wien, 1984.
2. E. G. Coffman, Jr., M. R. Garey, and D. S. Johnson. Approximation algorithms for bin packing: A survey. In D. Hochbaum (ed), PSW publishing, Boston. Approximation Algorithms for NP-Hard Problems, pages 46-93. 1996.
3. 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.
4. E. G. Coffman, Jr., C. A. 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. Discrete Math., vol. 13 pp. 384-402, 2000.
5. E. G. Coffman, Jr., D. S. Johnson, P. W. Shor and R. R. Weber. Bin packing with discrete item sizes, Part II: Tight bounds on First Fit. Random Structures and Algorithms, 10: 69-101, 1997.
6. E. G. Coffman, Jr., D. S. Johnson, P. W. Shor and R. R. Weber. Markov chains, computer proofs and average case analysis of Best Fit bin packing. In Proceedings of the 25th Annual ACM Symposium on Theory of Computing, pages 412-421. ACM Press, 1993.
7. E. G. Coffman Jr., Shlomo Halfin, Alain Jean-Marie and Philippe Robert. Stochastic analysis of a slotted FIFO communication channel. IEEE Trans. on Info. Theory, Vol. 39, No. 5, pp. 1555-1566, 1993.
8. E. G. Coffman Jr., Kimming So, Micha Hofri and A. C. Yao. A Stochastic model of bin packing. Information and Control, Vol. 44, pp. 105-115, 1980
9. E. G. Coffman, Jr. and G. S. Lueker. Probabilistic Analysis of Packing and Partitioning Algorithms. Wiley, New York, 1991
10. J. Csirik and D. S. Johnson. Bounded Space On Line Bin Packing: Best is better than First. Proc. of the 2nd ACM-SIAM Symp. on Discrete Algorithms, pp. 309-319, 1991.
11. J. Csirik, J.B.G. Frenk, A.M. Frieze, G. Galambos and A.H.G. Rinnoy Kan. A Probabilistic analysis of the Next-Fit Decreasing bin packing heuristic. Operations Research Letters, Vol. 5 No. 5 pp. 233-236, 1986.
12. D. K. Friesen and M. A. Langston. Analysis of a compound bin packing algorithm. SIAM J. Disc. Math, 4:61-79, 1991.
13. M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Co., San Francisco, 1979.
14. Micha Hofri. A Probabilistic analysis of the Next-Fit bin packing algorithm. Journal of Algorithms, Vol. 5, pp. 547-556, 1984.
15. M. Hofri. and S. Kamhi. A stochastic analysis of the NFD bin packing algorithm. Journal of Algorithms, Vol. 7, pp. 489-509, 1986.
16. N. Karmarkar. Probabilistic analysis of some bin packing algorithms. Proc. 23rd IEEE FOCS, pp. 107-111, 1982.
17. 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. of Computing, 3:299-325, 1974.
18. D. S. Johnson. Fast algorithms for bin packing. Journal of computer and system Science. Vol. 8, pp. 272-314, 1974.
19. D. S. Johnson, Near-Optimal Bin Packing Algorithms, Doctoral Dissertation, Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, 1973
20. N. Karmarkar and R. M. Karp. An efficient approximation scheme for the one-dimensional bin packing problem. In Proc. 23rd Ann. Symp on Foundations of Computer Science, pp. 312-320, 1982.
21. C. C. Lee and D. T. Lee. A simple on-line bin-packing algorithm. Journal Assoc. Comput. Mach., vol. 32, pp. 562-572, 1985.
22. C. C. Lee and D. T. Lee. Robust on-line bin-packing algorithms. Technical report, Department of Elct. Engineering and CS. Northwestern University, Evanston, IL, 1987.
23. W. Mao. Best-k-Fit bin packing. Computing, vol. 50, pp. 265-270, 1993.
24. J.R. Norris. Markov Chains. Cambridge University press, New-York. 1997.
25. P. Ramanan. Average case analysis of the smart Next Fit algorithm. Information Processing Letters, 31, pp. 221-225, 1989.
26. T. Rhee. Probabilistic analysis of the Next-Fit Decreasing algorithm for bin packing. Operations Research Letters, vol. 6 No. 4, pp. 189-191, 1987.
27. T. Rhee. and M. Talagrand. Martingale inequalities and NP-complete problems. Mathematical Operations Research , vol. 12, pp. 177-181, 1987.
28. G. Zhang. Tight worst-case performance bound for AFBK. Report No. 015, Institute of Applied Mathematics, Beijing, China, 1994.