Average Case Analysis of Bounded Space Bin Packing Algorithms

Nir Naaman and Raphael Rom

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

We analyze the average case performance of bounded space bin packing algorithms. The analysis is based on a novel 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 applications of bin packing. However, standard average case analysis assume that items are chosen from a continuous interval. We show that many important results are lost in the transition from the discrete to the continuous distribution. Our technique is general enough to calculate results for any discrete 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 relevance to a large number of applications and because of its theoretical significance bin packing has been widely researched and investigated (for an extensive survey see, [3]). 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 $s(a_i) \in (0, 1]$ and are asked to pack them into a minimum number of unit capacity bins. Since the problem is NP-hard [12], many approximation algorithms have been developed for it (see, [3] 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-bounded space* 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^∞ 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 can be too pessimistic, since the worst case may rarely occur. A different approach for estimating the performance of an algorithm is an average case analysis. In this case, we assume that item sizes 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 standard definition of the expected performance ratio is [3]

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

We are interested in the *asymptotic expected performance ratio* which is defined as

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

As will be shown, all the algorithms we consider have the property that $\lim_{n\to\infty} E\left[A(L_n)/n\right]$ exists under any discrete item size distribution H. It follows that the limit in (3) exists, hence $\overline{R}_A^\infty(H)$ is well defined (see details in subsection A.2).

In some cases we are not interested, or unable, to compare the performance of the algorithm to that of an optimal packing. Instead, we are interested in the expected bin utilization of the algorithm. Let U be the bin size and denote by $s(L_n)$ the total size of all items in L_n. We define the asymptotic expected bin utilization of algorithm A for distribution H, $\overline{\eta}_A^\infty(H)$, as

$$\overline{\eta}_A^\infty(H) \equiv \lim_{n\to\infty} E\left[\frac{s(L_n)/U}{A(L_n)}\right] \tag{4}$$

Similar to $\overline{R}_A^\infty(H)$, since we consider algorithms for which $\lim_{n\to\infty} E\left[A(L_n)/n\right]$ exists, $\overline{\eta}_A^\infty(H)$ is well defined.

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, since the results of an average case analysis depend on the item-size distribution, it is desirable to be able to calculate results for any given distribution. Although other cases have been studied, most of the results that have been published to date, concern cases where the items are independent, identically distributed (i.i.d.) with a uniform distribution [3]. The uniform distribution appears in two forms:

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

most early work on average case analysis of bin packing algorithms assumed a continuous distribution [3]. 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. [1]. In a series of papers, they considered algorithms such as First-Fit [6] and Best-Fit [5] as well as the optimal packing [2]. 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 exist only for the Next Fit algorithm [4]. Results for the rest of the algorithms are based on simulations [11].

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 conjectured from simulation. In Section 2 we present the technique of our average case analysis. In Section 3 and Section 4 we analyze the best known 1-bounded space and 2-bounded space algorithms, respectively. Section 5 explains 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 discrete item size distribution H, requiring only that item sizes be independent, identically distributed. We denote by h_i the probability that an item has 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 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. The probability distribution for N_{t+1} is completely determined by the value of N_t, which renders the process a Markov chain. 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 subset of states which are recurrent and accessible from the initial state (empty bins); we refer to these as the *repeated-states* of the Markov chain. Since the chain is finite such a subset always exists. We assume for now that the repeated-states are aperiodic and hence comprise an ergodic chain. In Appendix A we elaborate more on the structure of the Markov chain and show that the analysis can be applied to periodic repeated-states also.

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 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, generated according to distribution H, we define the expected average combined size of all items to be

$$I_A^n(H) \equiv E\left[\frac{1}{n}\sum_{t=1}^{n}(s(a_t) + oh_t)\right] \tag{5}$$

The expected *asymptotic average combined size* of all items is defined as

$$I_A(H) \equiv \lim_{n\to\infty} I_A^n(H) \tag{6}$$

The existence of the limit in (6) will become clear from the analysis.

The overhead added to all the items accounts for the wasted space in all but the last bin; hence, $U \cdot A(L_n) - \sum_{t=1}^{n}(s(a_t) + oh_t) < U$. We can therefore state the following relation between $I_A(H)$ and $A(L)$

$$I_A(H) = \lim_{n\to\infty} E\left[\frac{U \cdot A(L_n)}{n}\right] \tag{7}$$

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 [24], 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 [3]. Therefore the asymptotic expected performance ratio is given by

$$\begin{aligned}\overline{R}_A^\infty(H) &= \lim_{n\to\infty} E\left[\frac{A(L_n)}{OPT(L_n)}\right] = \lim_{n\to\infty} E\left[\frac{\frac{U}{n}A(L_n)}{\frac{U}{n}OPT(L_n)}\right] \\ &= \frac{\lim_{n\to\infty} E\left[\frac{U}{n}A(L_n)\right]}{\lim_{n\to\infty} E\left[\frac{U}{n}OPT(L_n)\right]} = \frac{I_A(H)}{I_{OPT}(H)}\end{aligned} \tag{8}$$

If we are only interested in the asymptotic expected bin utilization we use $s(L_n)/U$ instead of $OPT(L_n)$. We denote by $\overline{h}$ the average item size, i.e., $\overline{h} = \sum_{i=1}^{U} i \cdot h_i$. Similar to (8) we now have

$$\overline{\eta}_A^\infty(H) = \lim_{n\to\infty} E\left[\frac{s(L_n)/U}{A(L_n)}\right] = \frac{\lim_{n\to\infty} E\left[\frac{1}{n}s(L_n)\right]}{\lim_{n\to\infty} E\left[\frac{U}{n}A(L_n)\right]} = \frac{\overline{h}}{I_A(H)} \tag{9}$$

To find the asymptotic expected performance ratio of the NF algorithm, we must calculate both $I_{OPT}(H)$ and $I_{NF}(H)$. Our analysis does not provide a way to calculate $I_{OPT}(H)$, a task that can be difficult for certain item size distributions. Fortunately, we do know that for several important distributions, including the uniform distribution, the overhead (wasted space) of the optimal packing can be neglected [3] (see details in subsection 2.1 therein). For such distributions we have $I_{OPT}(H) = \overline{h}$. To find $I_{NF}(H)$ 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 list of n items; denote by V_j^n the number of visits in state j during the packing. Since we consider ergodic chains, we have $Pr\left(\lim_{n\to\infty} \frac{V_j^n}{n} = \Pi_j\right) = 1$, or in short $\lim_{n\to\infty} \frac{V_j^n}{n} = \Pi_j$, *a.s.* (almost surely).

We now denote by $V_{j,i}^n$ the number of items of size i which are 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. We can, therefore, use the law of large numbers to establish the following property of $V_{j,i}^n$:

$$\lim_{n\to\infty} \frac{V_{j,i}^n}{n} = \lim_{n\to\infty} \frac{V_j^n}{n} \cdot h_i = \Pi_j \cdot h_i\,, \ \ a.s. \tag{10}$$

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:

$$\begin{aligned}I_A(H) = \lim_{n\to\infty} I_A^n(H) &= \lim_{n\to\infty} E\left[\frac{1}{n}\sum_{j=1}^{U}\sum_{i=1}^{U} V_{j,i}^n \cdot (i + oh_i(j))\right] \\ &= E\left[\sum_{j=1}^{U}\sum_{i=1}^{U} \lim_{n\to\infty} \frac{V_{j,i}^n}{n} \cdot (i + oh_i(j))\right]\end{aligned} \tag{11}$$

Substituting (10) we get

$$I_A(H) = \sum_{j=1}^{U}\sum_{i=1}^{U} \Pi_j \cdot h_i \cdot (i + oh_i(j)) \tag{12}$$

To simplify (12) we use the following definitions:

$$\begin{aligned}OH(j) &\equiv \textstyle\sum_{i=1}^{U} h_i \cdot oh_i(j) \quad \text{average overhead in state } j \\ \overline{OH} &\equiv \textstyle\sum_{j=1}^{U} \Pi_j \cdot OH(j) \quad \text{average overhead size}\end{aligned} \tag{13}$$

Equation (12) now becomes

$$\begin{aligned} I_A(H) &= \sum_{j=1}^{U} \Pi_j \cdot \sum_{i=1}^{U} i \cdot h_i + \sum_{j=1}^{U} \Pi_j \cdot \sum_{i=1}^{U} h_i \cdot oh_i(j) \\ &= \overline{h} + \sum_{j=1}^{U} \Pi_j \cdot OH(j) = \overline{h} + \overline{OH} \end{aligned} \tag{14}$$

The expression in (14) is intuitive; the asymptotic average combined size of the items is the average size of the items plus the average size of the overhead.

The application of the technique to other algorithms is straightforward. We only describe the outlines here; a detailed analysis is presented for each algorithm in the next sections. For a K-bounded space algorithm, A_K, the state of the packing is the content of the K open bins $N_t = (j_1, j_2, ..., j_K)$, where $1 \le j_1, j_2, ..., j_K \le U$. If the state of the algorithm can be described by a finite Markov chain, we can apply the analysis. We use the rules of the algorithm to construct the transition matrix P and calculate the equilibrium probabilities $\Pi(j_1, j_2, ..., j_K)$. Next we calculate $oh_i(j_1, j_2, ..., j_K)$, the overhead which is added to an item of size i in each state. The average combined size of the items is calculated in the following way:

$$I_{A_K}(H) = \overline{h} + \sum_{j_1=1}^{U} \cdots \sum_{j_K=1}^{U} \sum_{i=1}^{U} h_i \cdot \Pi(j_1, ..., j_K) \cdot oh_i(j_1, ..., j_K) \tag{15}$$

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 discrete item size distribution. However, the discrete uniform distribution is of special interest and has been the focus of most previous work [3]. 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., $h_i = \frac{1}{U}$, $1 \le i \le U$.
2. Analysis of the distribution $\{u, U\}$, i.e., $h_i = \frac{1}{u}$, $1 \le i \le u$, $h_i = 0$, $u < i \le U$.

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 approach 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 independently 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

$$\overline{W}_A^n(H) = E\left[U \cdot A(L_n) - s(L_n)\right] \tag{16}$$

Distribution H is said to allow *perfect packing* if $\overline{W}_{OPT}^n(H) = o(n)$. For such distributions we may neglect the overhead when calculating $I_{OPT}(H)$. Ignoring the last bin we have

$$\begin{aligned} I_{OPT}(H) &\equiv \lim_{n\to\infty} E\left[\frac{1}{n}\sum_{t=1}^{n}(s(a_t) + oh_t)\right] = \lim_{n\to\infty} E\left[\frac{1}{n}\left(\overline{W}_{OPT}^n + s(L_n)\right)\right] \\ &= \lim_{n\to\infty} E\left[\frac{1}{n}\left(o(n) + s(L_n)\right)\right] = \lim_{n\to\infty} E\left[\frac{1}{n}s(L_n)\right] = \overline{h} \end{aligned} \tag{17}$$

Several studies have tried to identify the type of distributions that allow perfect packing [7, 3]. Necessary and sufficient conditions for a given discrete distribution to allow perfect packing are described in [9]; analogous results for continuous distributions are given in [25]. The discrete uniform distribution $\{u, U\}$ allows perfect packing, for any values of u and U, since its wasted space is the following [2]

$$\overline{W}^n_{OPT}(\{u, U\}) = \begin{cases} O(1) & u < U - 1 \\ \Theta\left(\sqrt{n}\right) & u \in \{U - 1, U\} \end{cases} \tag{18}$$

Based on the above results, we conclude that for the distribution $\{u, U\}$

$$I_{OPT}(\{u, U\}) = \sum_{i=1}^{u} i \cdot h_i = \frac{1}{u} \sum_{i=1}^{u} i = \frac{u+1}{2} \tag{19}$$

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; hence the repeated-states form an ergodic chain. 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 standard 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 given general distribution H, we denote by h_i the probability of an item being of size i. Our only assumption is that the items are i.i.d. For a given algorithm A and distribution H our analysis enables us to calculate $I_A(H)$ which means we can find the expected bin utilization of the algorithm. If we are interested in the asymptotic expected performance ratio we must also find $I_{OPT}(H)$. However, as we pointed out, finding $I_{OPT}(H)$ for certain item size distributions may not be easy.

In cases where $I_{OPT}(H)$ is not known we can still get meaningful results by calculating the expected bin utilization $\overline{\eta}^\infty_A$ using (9). Note that $(\overline{\eta}^\infty_A)^{-1}$ serves as an upper bound on the performance ratio, since $s(L)/U \le OPT(L)$. For example, if all items are of size $3U/4$ the average combined size (of any algorithm) is U and the bin utilization is $\overline{\eta}^\infty_A = 3/4$. However in this case the performance ratio is 1, since the optimal packing can not produce a better packing.

We use (14) to calculate the average combined size of the items. We must find two components:

1. The equilibrium probabilities of the Markov chain, Π.
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 construct the transition matrix P and solve the set of equations $\Pi = \Pi P$. The calculation of the overhead component $oh_i(j)$, for all the algorithms we consider, is also simple. Calculating the average combined size of the items is, therefore, 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, subsection 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 algorithms and then summarize the results in subsection 3.4. While Next Fit and Smart Next Fit are online algorithm with $O(n)$ running time, Next Fit Decreasing is an offline 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 $\overline{R}^{\infty}_{NF}([0,1]) = \frac{4}{3}$. Results for the $[0,b]$ distribution, where $0 < b \leq 1$, have been reported by Karmarkar in [17]. The only results for discrete item sizes are for the $\{U,U\}$ distribution. It has been shown in [4] that the NF algorithm has the following asymptotic expected performance ratio:

$$\overline{R}^{\infty}_{NF}(\{U,U\}) = \frac{2(2U+1)}{3(U+1)} \tag{20}$$

As we expect, 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, 17, 13]). 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 $j \leq \left\lfloor \frac{U}{2} \right\rfloor$ we have

$$P_{j,k} = \frac{1}{U} \cdot \begin{cases} 0 & 1 \leq k \leq j \\ 1 & j < k \leq U-j \\ 2 & U-j < k \leq U \end{cases} \quad \Bigg| \quad 1 \leq j \leq \left\lfloor \frac{U}{2} \right\rfloor \tag{21}$$

The last line is $P_{U,k} = \frac{1}{U}, \;\; 1 \leq k \leq U$.

The simple structure of the matrix P enables an easy solution to the set of equations $\Pi = \Pi P$.

$$\Pi_j = \frac{2j}{U\,(U+1)} \tag{22}$$

Next we compute the overhead component $OH(j)$. When NF is in state j any item bigger than $U-j$ creates an overhead of $U-j$ units. Hence, the average overhead in state j is

$$OH(j) = \sum_{i=1}^{U} h_i \cdot oh_i(j) = \sum_{i=U-j+1}^{U} \frac{1}{U} \cdot (U-j) = \frac{j(U-j)}{U} \tag{23}$$

We now use (22) and (23) to find the average combined size of the items

$$\begin{aligned} I_{NF}(\{U,U\}) &= \frac{U+1}{2} + \sum_{j=1}^{U} \Pi_j \cdot OH(j) = \frac{U+1}{2} + \sum_{j=1}^{U} \frac{2j}{U(U+1)} \cdot \frac{j(U-j)}{U} \\ &= \frac{U+1}{2} + \sum_{j=1}^{U} \frac{2j^2\,(U-j)}{U^2\,(U+1)} = \frac{U+1}{2} + \frac{U-1}{6} = \frac{2U+1}{3} \end{aligned} \tag{24}$$

We use $I_{NF}(\{U,U\})$ and $I_{OPT}(\{U,U\})$ to obtain the asymptotic expected performance ratio

$$\overline{R}^{\infty}_{NF}(\{U,U\}) = \frac{I_{NF}(\{U,U\})}{I_{OPT}(\{U,U\})} = \frac{(2U+1)/3}{(U+1)/2} = \frac{2(2U+1)}{3(U+1)} \tag{25}$$

Our result is in accordance with the one reported in [4].

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

To construct the transition matrix 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, $1 \leq i \leq u$. We distinguish between two cases:

1. When $j + i \leq U$ the item fits in the open bin and the next state is $N_t = j + i$.
2. When $j + i > U$ the item does not fit in the open bin; it is therefore packed in a new bin 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 $\Pi = \Pi P$. Calculating Π numerically is easy. However, in order to get a better understanding of the results, we show in Appendix B how to derive a closed form of Π.

Our next step is to calculate $OH(j)$. Assume that the algorithm is in state $N = j$ and the next item to be packed is of size i, $1 \leq i \leq u$. 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

$$OH(j) = \begin{cases} 0 & j \leq U - u \\ \frac{(j+u-U)\cdot(U-j)}{u} & j > U - u \end{cases} \tag{26}$$

Once we have the equilibrium probabilities Π_j and the average overhead in state j, $OH(j)$, we use (14) to find the average combined size of the items, $I_{NF}(\{u,U\})$. The asymptotic expected performance ratio is then calculated from the following expression:

$$\begin{aligned} \overline{R}^{\infty}_{NF}(\{u,U\}) &= \frac{I_{NF}(\{u,U\})}{I_{OPT}(\{u,U\})} = 1 + \frac{2}{u+1}\sum_{j=1}^{U} \Pi_j \cdot OH(j) \\ &= 1 + \frac{2}{u+1} \sum_{j=U-u+1}^{U} \Pi_j \cdot \frac{(j+u-U)\,(U-j)}{u} \end{aligned} \tag{27}$$

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 [17]. We present some computational results for the NF algorithm in Section 3.4.

It is possible to obtain a closed form of $\overline{R}^{\infty}_{NF}(\{u,U\})$; however, the expression gets very complex unless u is very small or very close to U. In Appendix B we show how to obtain a closed form of $\overline{R}^{\infty}_{NF}(\{u,U\})$ and derive such closed form for certain value of u. We also derive a very good approximation which is more practical (see Appendix B)

$$\overline{R}^{\infty}_{NF}(\{u,U\}) \approx \begin{cases} \frac{3U}{3U-u+1} & u < \frac{U}{2} \\ \frac{3U^3-3(5u+1)U^2+u(23u+10)U-3u^2(u+1)}{6u^2(u+1)} & \frac{U}{2} < u \leq U \end{cases} \tag{28}$$

The approximation error is less than 10^{-3} when u is not too close to $\frac{U}{2}$. As u becomes closer to $\frac{U}{2}$ the approximation error increases to about 10^{-2}. In Figure 1 we compare the exact values of $\overline{R}_{NF}^{\infty}(\{u,U\})$ with the approximation we obtain by using (28). The comparison is for $U = 50$; the approximation error in this case is no more than 0.01.

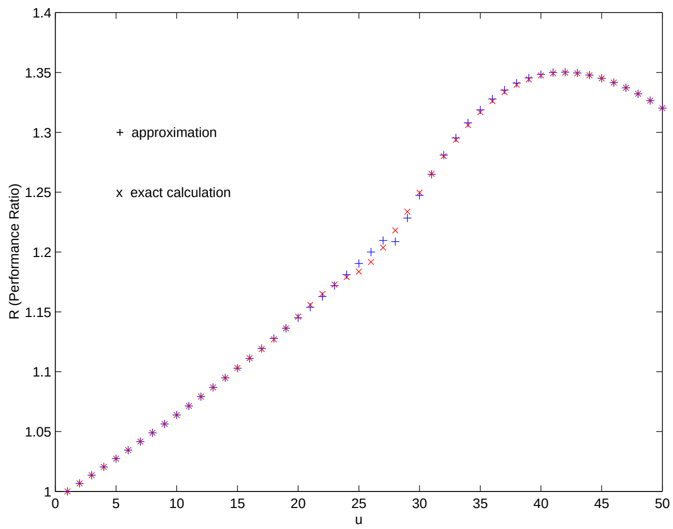

FIG. 1. Approximation of $\overline{R}_{NF}^{\infty}(\{u,U\})$ vs. the exact calculation for distribution $\{u,U\}$ for $U = 50$.

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 finding $I_{OPT}(H)$ may be difficult, we calculate the bin utilization which requires finding $I_{NF}(H)$ only. We use (14) 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

$$oh_i(j) = \begin{cases} 0 & j+i \leq U \\ U-j & j+i > U \end{cases} \tag{29}$$

Example: Consider the case of a communication channel in which variable size datagrams are transmitted in fixed-size packets. We assume a distribution $\tilde{H}$ of common Ethernet datagram sizes and probabilities; 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. We are interested in evaluating the expected channel utilization. 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 $I_{NF}(\tilde{H}) = 455.5$ while the average item size is $\overline{h} = 320$. The bin utilization is therefore $\overline{\eta}_{NF}^{\infty}(\tilde{H}) = 0.702$; clearly this is also the channel utilization. It is easy to verify that in this example a packing with $O(1)$ wasted space exists. Therefore, $I_{OPT}(\tilde{H}) = \overline{h} = 320$ and the performance ratio of the algorithm is $\overline{R}_{NF}^{\infty}(\tilde{H}) = 1.423$. It is interesting to note that the performance ratio is considerably worse (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 [22]. 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. Our work, to the best of our knowledge, is the first analysis of SNF for discrete item size distribution.

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 $\left\lfloor \frac{U}{2} \right\rfloor$ in all the equations.

We use (14) to find the average combined size of the items. The transition matrix P is

$$P_{j,k} = \frac{1}{U} \cdot \left\{ \begin{array}{cl} 0 & 2 \le j \le U/2\,,\; k \le j-1 \\ 0 & U/2+1 \le j < U\,,\; k \le U-j \\ j & j = k\,,\; j \le U/2 \\ U+1-j & j = k\,,\; U/2+1 \le j \le U \\ 1 & else \end{array} \right| \; 1 \le j \le U \tag{30}$$

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

$$\Pi_j = \left\{ \begin{array}{cl} \frac{j}{(U-j)(U-j+1)} & 1 \le j \le U/2 \\ \frac{1}{j} & U/2+1 \le j \le U \end{array} \right. \tag{31}$$

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, depending 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.

$$OH(j) = \sum_{i=1}^{U} \frac{1}{U} \cdot oh_i(j) = \frac{1}{U} \sum_{i=U-j+1}^{U} (U-i) = \frac{j(j-1)}{2U}\,, \quad 1 \le j \le U/2 \tag{32}$$

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.

$$OH(j) = \frac{1}{U} \sum_{i=U-j+1}^{j} (U-j) + \frac{1}{U} \sum_{i=j+1}^{U} (U-i) = \frac{(U-j)(3j-U-1)}{2U}\,, \quad U/2 < j \le U \tag{33}$$

We can now calculate the combined average size of the items

$$I_{SNF}(\{U,U\}) = \frac{U+1}{2} + \sum_{j=1}^{U/2} \frac{j}{(U-j)(U-j+1)} \cdot \frac{j(j-1)}{2U} + \sum_{j=U/2+1}^{U} \frac{1}{j} \cdot \frac{(U-j)(3j-U-1)}{2U} \tag{34}$$

We can obtain a closed form of (34) using Harmonic numbers $H_n = \sum_{i=1}^{n} \frac{1}{i}$ (the calculation is presented in Appendix C).

$$I_{SNF}(\{U,U\}) = \begin{cases} 2U - (2U+1)(H_U - H_{U/2}) & U \text{ is even} \\ \frac{2U^2}{U+1} - (2U+1)(H_U - H_{\lceil U/2 \rceil}) & U \text{ is odd} \end{cases} \tag{35}$$

The asymptotic expected performance ratio of SNF is therefore

$$\overline{R}^{\infty}_{SNF}(\{U,U\}) = \frac{I_{SNF}(\{U,U\})}{(U+1)/2} = \begin{cases} \frac{4U}{U+1} - \frac{2(2U+1)}{U+1}(H_U - H_{U/2}) & U \text{ is even} \\ (\frac{2U}{U+1})^2 - \frac{2(2U+1)}{U+1}(H_U - H_{\lceil U/2 \rceil}) & U \text{ is odd} \end{cases} \tag{36}$$

Using the approximation $H_n \approx \ln(n)$ we get

$$\overline{R}^{\infty}_{SNF}(\{U,U\}) \approx \frac{4U}{U+1}(1-\ln 2) \approx 1.227 \frac{U}{U+1} \tag{37}$$

When $U \to \infty$ we approach the uniform continuous distribution. Our result match the one reported in [22], $\overline{R}^{\infty}_{SNF}([0,1]) = 1.227\ldots$.

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

To calculate the equilibrium probabilities assume that at time $t-1$ SNF 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 construct the transition matrix P and find the equilibrium probabilities. Calculating Π numerically is straightforward but it is also possible to obtain a closed form of the equilibrium probabilities. Unfortunately, this closed form is quite complex for most values of u. In Appendix C we elaborate on how a closed form expression can be derived.

Our next step is to calculate the overhead. Assume that SNF is in state j and the next item to be packed is of size i, $1 \le i \le u$. Overhead units are added only if $i > U - j$, in which case there are two possibilities: 1) When $i \le j$ the open bin is changed and the overhead is $U - j$, and 2) When $i > j$ the item is packed in a new bin which is immediately closed and the overhead is $U - i$.

$$oh_i(j) = \begin{cases} 0 & j+i \le U \\ U-j & j+i > U,\ j \ge i \\ U-i & j+i > U,\ j < i \end{cases} \tag{38}$$

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

$$OH(j) = \frac{1}{u} \cdot \begin{cases} 0 & j \leq U-u \\ \sum_{i=U-j+1}^{u} (U-i) & U-u < j \leq U/2 \\ (min\{u,j\} - U + j)(U-j) + \sum_{i=j+1}^{u} (U-i) & max\{U-u,\ U/2\} < j \end{cases} \tag{39}$$

Once we have Π and $OH(j)$ we use (14) to find $I_{SNF}(\{u,U\})$. The asymptotic expected performance ratio is then calculated from the following expression:

$$\overline{R}_{SNF}^{\infty}(\{u,U\}) = \frac{I_{SNF}(\{u,U\})}{I_{OPT}(\{u,U\})} = 1 + \frac{2}{u+1} \sum_{j=1}^{U} \Pi_j \cdot OH(j) \tag{40}$$

We present some computational results for the SNF algorithm in Section 3.4. The results corresponding to the continuous case, i.e., when $U \to \infty$, match the results reported in [22].

A closed form expression of the expected performance ratio is very complex for most values of u. Instead we provide a very good approximation of $\overline{R}_{SNF}^{\infty}(\{u,U\})$ (see details in Appendix C). Since SNF is identical to NF when $u \leq \frac{U}{2}$ we use the approximation of NF in this range (see (28)). For $u > \frac{U}{2}$ we have the following approximation:

$$\begin{aligned} \overline{R}_{SNF}^{\infty}(\{u,U\}) \approx\ & \frac{(u^2(7u^2+1)(7u+5) - (6u+8u^2(4u+3))U + (1+6u(13u+2))U^2}{6Uu^2(u+1)} \\ & - \frac{(28u+U)U^3)}{6Uu^2(u+1)} - \frac{2(2U+1)U(H_u - H_{U/2})}{u(u+1)} \quad , \quad u > U/2 \end{aligned} \tag{41}$$

Finally we note that calculating result for other item size distributions is similar to the calculation of the $\{u,U\}$ distribution. We first construct the transition matrix and calculate the equilibrium probabilities. We then use (38) as the overhead component and calculate $I_{SNF}(H)$ using (14).

$$I_{SNF}(H) = \overline{h} + \sum_{j=1}^{U} \Pi_j \cdot \sum_{i=1}^{U} h_i \cdot oh_i(j) \tag{42}$$

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 shall 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, have been presented in [10], [14] and [23]. All previous work assumed the $[0,1]$ continuous uniform distribution. It has been shown that the asymptotic expected performance ratio of the algorithm is $\overline{R}_{NFD}^{\infty}([0,1]) = 2\left(\frac{\pi^2}{6} - 1\right) = 1.289\ldots$. Our work, to the best of our knowledge, is the first analysis of NFD for discrete item size distribution.

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

We are interested in the average combined size of the items. However, this time 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. Note that the number of bins that contain items of more than one size is at most $U/2$, so when considering long lists ($n \gg U$) such bins can be ignored.

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. Their combined size is therefore U.
2. The average combined size of items of size $i \le U/2$ is $U/\lfloor U/i \rfloor$.

The asymptotic average combined size of the items is

$$I_{NFD}(\{U,U\}) = \sum_{i=1}^{U/2} \frac{1}{U} \cdot \frac{U}{\lfloor U/i \rfloor} + \sum_{i=U/2+1}^{U} \frac{1}{U} \cdot U = \frac{U}{2} + \sum_{i=1}^{U/2} (\lfloor U/i \rfloor)^{-1} \tag{43}$$

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

$$\overline{R}^{\infty}_{NFD}(\{U,U\}) = \frac{U}{U+1} + \frac{2}{U+1} \sum_{i=1}^{U/2} (\lfloor U/i \rfloor)^{-1} \tag{44}$$

We present some computational results for the NFD algorithm in Section 3.4. When $U \to \infty$ we get a result which is in agreement with the value of the asymptotic expected performance ratio of NFD for the uniform continuous distribution [0,1] [10].

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 (see Figure 2). 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.

Since the asymptotic expected performance ratio of NFD is oscillating, it is difficult to express it in closed form. It is easy, however, to derive a lower bound on $\overline{R}^{\infty}_{NFD}(\{U,U\})$ by ignoring the floor function in (44). By slightly modifying the lower bound we obtain an upper bound.

$$\frac{5U+2}{4(U+1)} \le \overline{R}^{\infty}_{NFD}(\{U,U\}) < \frac{5.2U+4}{4(U+1)} \tag{45}$$

The lower bound equals the exact value only for $U \in \{4,6\}$. Asymptotically the lower bound is 1.25 and the upper bound is 1.3, while the exact value is $1.289\ldots$.

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 (43) becomes

$$I_{NFD}(\{u,U\}) = \sum_{i=1}^{min\{U/2,\,u\}} \frac{1}{u} \cdot \frac{U}{\lfloor U/i \rfloor} + \sum_{i=U/2+1}^{u} \frac{1}{u} \cdot U \tag{46}$$

Note that the second term in (46) is zero for any $u \le U/2$.

The extension of the analysis to a general distribution is immediate. We should only change (43) to give a proper weight to each item size.

$$I_{NFD}(H) = \sum_{i=1}^{U/2} h_i \cdot \frac{U}{\lfloor U/i \rfloor} + \sum_{i=U/2+1}^{U} h_i \cdot U \tag{47}$$

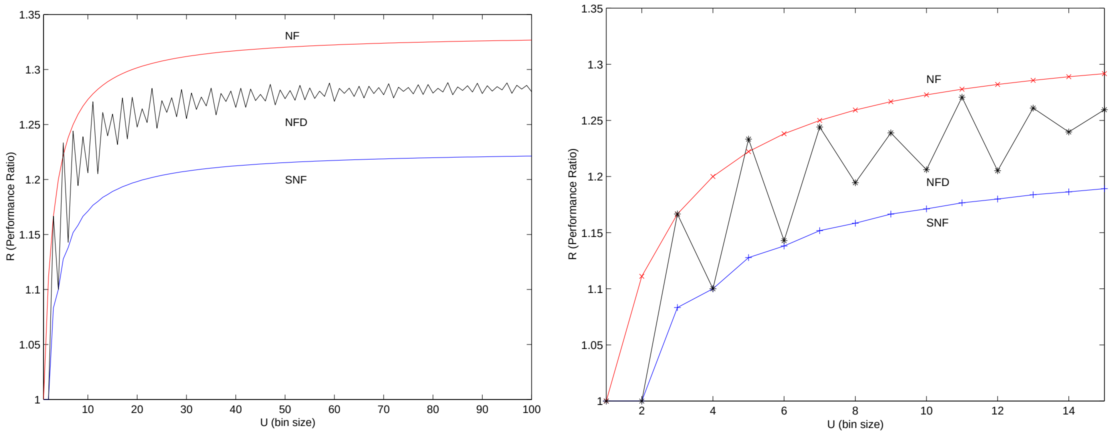

FIG. 2. Asymptotic expected performance ratio for distribution $\{U, U\}$.

3.4. Summary of Results

We present some results of our average case analysis, for several values of U, in Table 1 (see Appendix D for details regarding the numerical calculations).

TABLE 1.

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

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

In Figure 2 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 monotonically 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.

Figure 3 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. Since NF and SNF are identical when $u \leq U/2$ their performance ratio is the same for these values. 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 [11]. 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.

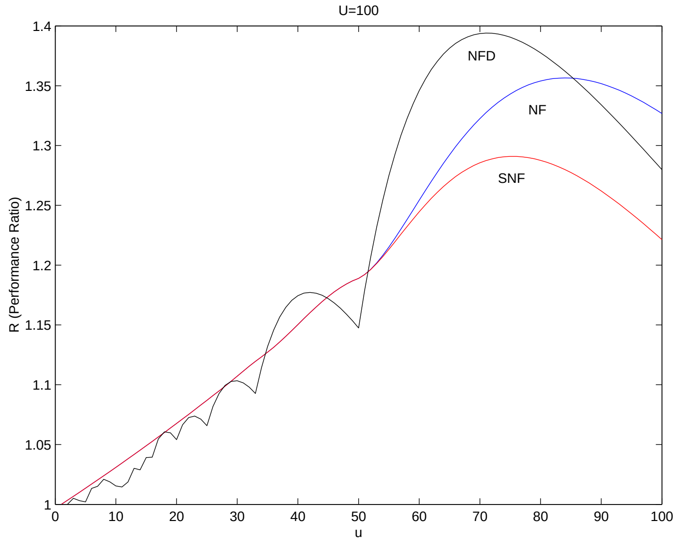

FIG. 3. Asymptotic expected performance ratio for distribution $\{u, U\}$ for $U = 100$.

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.

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

We add one more closing rule, which we call 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 the closing rule is applied (assuming there are already K open bins) 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 that have been studied. One important class we do not consider here are the *Harmonic* algorithms H_K [18], for which an average case analysis is relatively easy and has been reported in [19].

In this section we present the analysis for the $\{u, U\}$ distribution and explain how to extend the analysis for a general distribution. 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 [16, 15]. Csirik and Johnson presented average case results based on simulation for different values of K in [11]. 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 $1 \leq j_1,\ j_2 \leq U$.

To construct the transition matrix P 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. 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 Π.

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, $1 \leq i \leq U$. Note that overhead units are added only if the next item does not fit in B_1 or B_2, that is, $i > U - min\{j_1, j_2\}$. When overhead units are added the overhead is the unused space in B_1, which is $U - j_1$.

$$oh_i(j_1, j_2) = \begin{cases} 0 & min\{j_1, j_2\} + i \leq U \\ U - j_1 & min\{j_1, j_2\} + i > U \end{cases} \tag{48}$$

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

$$OH(j_1, j_2) = max\left\{\frac{(u + min\{j_1, j_2\} - U) \cdot (U - j_1)}{u},\ 0\right\} \tag{49}$$

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

$$\overline{R}^{\infty}_{NF_2}(\{u, U\}) = \frac{I_{NF_2}(\{u, U\})}{I_{OPT}(\{u, U\})} = 1 + \frac{2}{u+1} \sum_{j_1=1}^{U} \sum_{j_2=1}^{U} \Pi(j_1, j_2) \cdot OH(j_1, j_2) \tag{50}$$

To calculate the average combined size of the items for a general distribution we use the equilibrium probabilities and (48) as the expression for the overhead component.

$$I_{NF_2}(H) = \sum_{j_1=1}^{U} \sum_{j_2=1}^{U} \sum_{i=1}^{U} h_i \cdot \Pi(j_1, j_2) \cdot (i + oh_i(j_1, j_2)) \tag{51}$$

We present some computational results for the 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 [11]. 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 [11]. 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 $1 \le j_2 \le j_1 \le U$, which means the number of states is $U(U+1)/2$.

To calculate the equilibrium 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. If $j_1 \ge j_2 + i$ 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 $j_2 \ge i$ 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 construct the transition matrix P and calculate the equilibrium probability vector Π.

We now calculate $OH(j_1, j_2)$. Overhead units are added only if the next item, of size i, 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, i.e., $U - j_1$.

$$oh_i(j_1, j_2) = \begin{cases} 0 & j_2 + i \le U \\ U - j_1 & j_2 + i > U \end{cases} \tag{52}$$

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

$$OH(j_1, j_2) = max\left\{\frac{(u + j_2 - U)\cdot(U - j_1)}{u},\ 0\right\} \tag{53}$$

We now use $\Pi(j_1, j_2)$ and $OH(j_1, j_2)$ to calculate the asymptotic expected performance ratio similar to (50). We use (51) 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

We introduce the Smart 2-Bounded Best Fit ($SBBF_2$) algorithm. $SBBF_2$ is similar to BBF_2 but includes the same improvement that Smart Next Fit has compered to 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 equilibrium 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 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, $1 \leq i \leq U$. 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, $U - max\{j_1, i\}$.

$$oh_i(j_1, j_2) = \begin{cases} 0 & j_2 + i \leq U \\ U - max\{j_1, i\} & j_2 + i > U \end{cases} \tag{54}$$

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

$$OH(j_1, j_2) = \begin{cases} 0 & j_2 + u \leq U \\ \frac{1}{u}\sum_{i=U-j_2+1}^{u} U - max\{j_1, i\} & j_2 + u > U \end{cases} \tag{55}$$

We now use $\Pi(j_1, j_2)$ and $OH(j_1, j_2)$ to calculate the asymptotic expected performance ratio similar to (50). We use (51) 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 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 $K \geq 2$ [20]. 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 [26]). 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 asymptotic expected performance ratio of 2-bounded space algorithms. See Appendix D for some details regarding the numerical calculations. 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

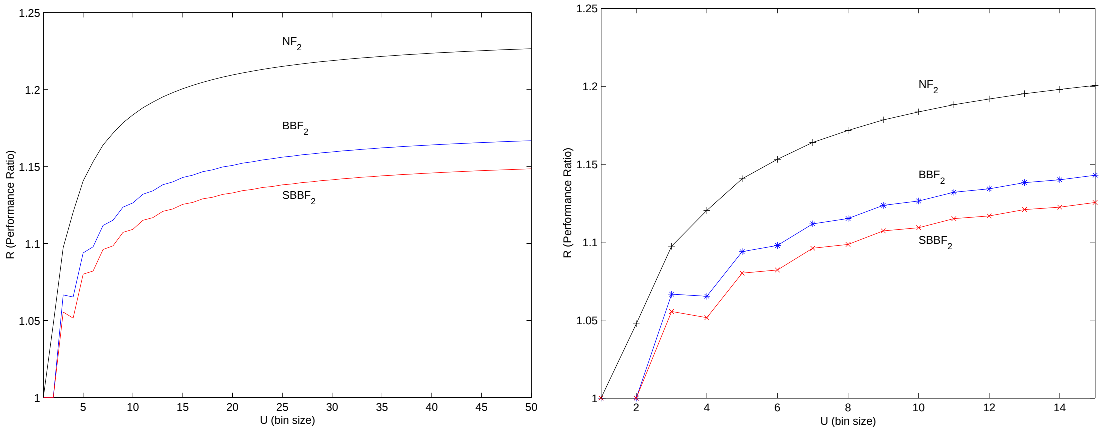

FIG. 4. Asymptotic expected performance ratio for distribution $\{U, U\}$.

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 [11].

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
∞	1.2386	1.1783	1.1600	1.2362	1.1926

Figure 4 present the expected performance ratio of the algorithms under the $\{U, U\}$ distributions for all values of $U \leq 50$. As a rule we can say that the expected 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.

Figure 5 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 BBF_2 is better than NF_2 for all values of u. Combining this observation with the results for the $\{U, U\}$ distribution (Figure 4) we conclude that Best-Fit is superior for any $\{u, U\}$ distribution. In Figure 6 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 $u = 0.85U$.

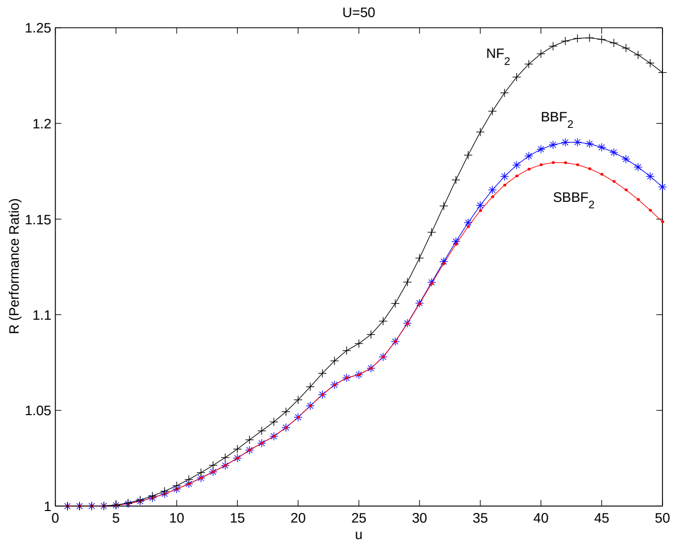

FIG. 5. Asymptotic expected performance ratio for distribution $\{u, U\}$ for $U = 50$.

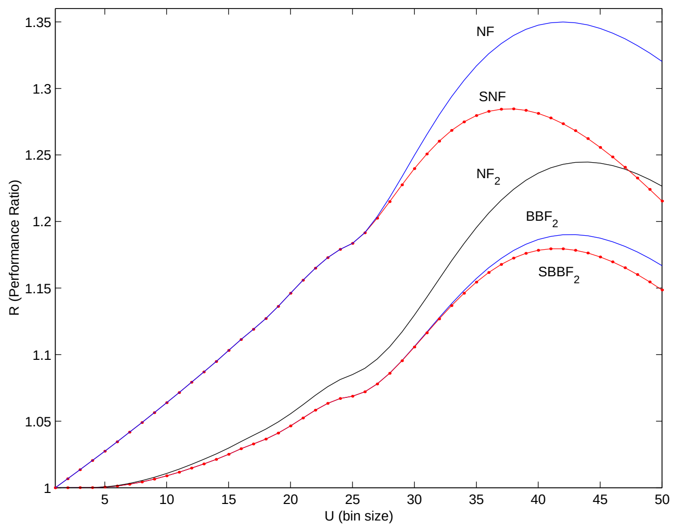

FIG. 6. Comparison of some bounded space algorithms for distribution $\{u, U\}$ for $U = 50$.

As an example for results of non uniform distributions we go back to the communication channel we presented in subsection 3.1.3. We found that the channel utilization of the NF algorithm is $\overline{\eta}^{\infty}_{NF}(\tilde{H}) = 0.70$. In order to calculate the channel utilization for our 2-bounded space algorithms we use (51). For NF_2 we find that adding an extra bin considerably improves the channel utilization $\overline{\eta}^{\infty}_{NF_2}(\tilde{H}) = 0.85$. The results for BBF_2 are even more surprising; the channel utilization is almost perfect $\overline{\eta}^{\infty}_{BBF_2}(\tilde{H}) = 0.998$.

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 similar. The state of the packing is the content of the K

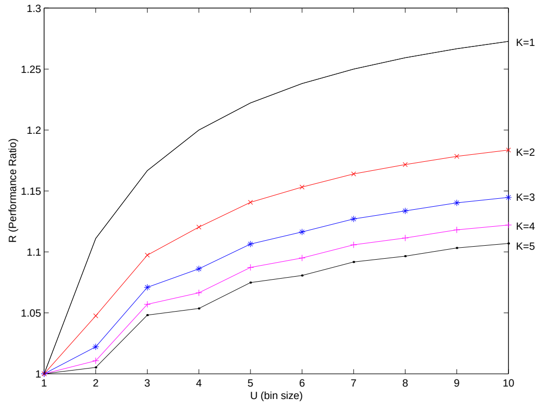

FIG. 7. Next-K Fit Performance ratio for values of $K \leq 5$, distribution $\{U, U\}$.

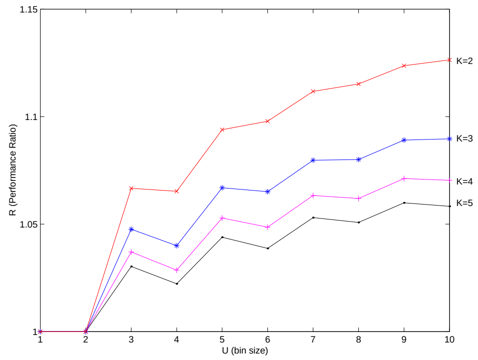

FIG. 8. K-Bounded Best Fit Performance ratio for values of $2 \leq K \leq 5$, distribution $\{U, U\}$.

open bins $N_t = (j_1, j_2, ..., j_K)$, where $1 \leq j_1, j_2, ..., j_K \leq U$. To calculate the asymptotic 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 $\Pi(j_1, j_2, ..., j_K)$ satisfying $\Pi = \Pi P$. Next we calculate the overhead component $oh_i(j_1, j_2, ..., j_K)$, i.e., the overhead which is added to an item of size i when it is packed in a given state. Finally we calculate the average combined size of the items in the following way:

$$I_{A_K}(H) = \sum_{j_1=1}^{U} \cdots \sum_{j_K=1}^{U} \sum_{i=1}^{U} h_i \cdot \Pi(j_1, ..., j_K) \cdot (i + oh_i(j_1, ..., j_K)) \tag{56}$$

We calculate the asymptotic expected performance ratio as

$$\overline{R}_A^{\infty}(H) = \frac{I_{A_K}(H)}{I_{OPT}(H)} \tag{57}$$

We can apply the same technique to analyze any algorithm for which the content of the open bins can be described by a finite Markov chain. We show, in Appendix A, that for such algorithms the equilibrium probabilities exists and I_{A_K} is well defined.

Figures 7 and 8 show 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 $U - 1$.

6. CONCLUDING REMARKS

In this paper we presented an average case analysis several bounded space bin packing algorithms. The analysis is based on a novel technique of average case analysis in which the asymptotic expected performance ratio of an algorithm is derived from the average combined size of the items. The packing of the algorithm is modelled 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 it is easy to calculate. The main drawback of the analysis lies in its computational complexity for those cases where a closed form cannot be derived. The number of possible states of the Markov chain increases as $O\left(U^K\right)$, which renders numerical calculations 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 asymptotic expected performance ratio for higher values of K is a subject we leave for future research.

REFERENCES

1. 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.
2. 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 Journal on Discrete Mathematics*, 13:384–402, September 2000.
3. E G Coffman Jr., M R Garey, and D S Johnson. Approximation algorithms for bin packing: A survey. In D Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*, pages 46–93. PSW, Boston, 1996.
4. E G Coffman Jr., S Halfin, A Jean-Marie, and P Robert. Stochastic analysis of a slotted FIFO communication channel. *IEEE Transactions on Information Theory*, 39(5):1555–1566, 1993.
5. 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.
6. 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.
7. E G Coffman Jr. and G S Lueker. *Probabilistic Analysis of Packing and Partitioning Algorithms.* Wiley, New York, 1991.
8. E G Coffman Jr., K So, and M Hofri. A stochastic model of bin packing. *Information and Control*, 44:105–115, 1980.
9. C A Courcoubetis and R R Weber. Necessary and sufficient conditions for stability of a bin packing system. *Journal of Applied Probability*, 23:989–999, 1986.
10. 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*, 5(5):233–236, 1986.
11. J Csirik and D S Johnson. Bounded space on line bin packing: Best is better than first. *Algorithmica*, 31(2):115–138, 2001.
12. 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.
13. M Hofri. A probabilistic analysis of the Next-Fit bin packing algorithm. *Journal of Algorithms*, 5:547–556, 1984.
14. M Hofri and S Kamhi. A stochastic analysis of the NFD bin packing algorithm. *Journal of Algorithms*, 7:489–509, 1986.
15. D S Johnson. *Near-Optimal Bin Packing Algorithms.* Doctoral dissertation, Mathematics, Massachusetts Institute of Technology, Cambridge, MA, 1973.
16. D S Johnson. Fast algorithms for bin packing. *Journal of computer and system Science*, 8:272–314, 1974.
17. N Karmarkar. Probabilistic analysis of some bin packing algorithms. In *Proceedings 23rd Annual Symposium on Foundations of Computer Science*, pages 107–111, 1982.
18. C C Lee and D T Lee. A simple on-line packing algorithm. *Journal of ACM*, 32:562–572, 1985.
19. C C Lee and D T Lee. Robust on-line bin-packing algorithms. Technical report, Department of Electrical Engineering and CS. Northwestern University, Evanston, IL, 1987.
20. W Mao. Best-k-Fit bin packing. *Computing*, 50:265–270, 1993.
21. J R Norris. *Markov Chains.* Cambridge University press, New York, 1997.
22. P Ramanan. Average case analysis of the smart next fit algorithm. *Information Processing Letters*, 31:221–225, 1989.
23. T Rhee. Probabilistic analysis of the next-fit decreasing algorithm for bin packing. *Operations Research Letters*, 6(4):189–191, 1987.

24. T Rhee and M Talagrand. Martingale inequalities and NP-complete problems. *Mathematical Operations Research*, 12:177–181, 1987.

25. T Rhee and M Talagrand. Optimal bin packing with items of random sizes. *SIAM J. on Computing*, 18:139–151, 1989.

26. G Zhang. Tight worst-case performance bound for AFBK. Technical Report #015, Institute of Applied Mathematics, Beijing, China, 1994.

APPENDIX A

A.1. EQUILIBRIUM PROBABILITIES

A.1.1. Existence of Equilibrium Probabilities

Our average case analysis is based on modelling the packing of an algorithm by a Markov chain. When the bin size is U and the algorithm is K-bounded space the number of possible states is U^K; the chain is therefore finite for every finite U and K. The relevant states to our analysis are only the subset of states which are recurrent and accessible from the initial state (empty bins); we refer to these states as the *repeated-states* of the Markov chain. Note that since the chain is finite such a subset always exists. States which are not repeated-states are either transient, or belong to a different irreducible class which is not accessible from the initial state (and are therefore never visited during the packing). Note that, due to the packing rules of the algorithms we consider, transient states may be visited only once.

Throughout the analysis we assume the existence of equilibrium probabilities and denote by Π_j the equilibrium probability of state j. We use these probabilities to evaluate the number of times the algorithm visits each state. Recall that we denote by V_j^n the number of visits in state j when the algorithm packs n items. In order for our analysis to work it is sufficient that the equilibrium probabilities satisfy the following property:

$$\Pi_j = \lim_{n\to\infty} \frac{V_j^n}{n}\ , \quad a.s. \quad \forall j. \tag{A.1}$$

We now argue that equilibrium probabilities with the above property exist and are unique for all the algorithms we considered, under any discrete item size distribution. 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.

THEOREM A.1. *Let H be any i.i.d. discrete item size distribution. The repeated-states of the Markov chain describing the packing of algorithm A_K, under distribution H, have unique equilibrium probabilities satisfying* $\Pi_j = \lim_{n\to\infty} \frac{V_j^n}{n}\ , \quad a.s. \quad \forall j.$

Proof. The repeated-states are the subset of states of the Markov chain which are recurrent and accessible from the initial state. Since we consider finite Markov chains, the repeated-states must all belong to the same irreducible class of the Markov chain, and are all positive recurrent. The repeated-states form a new Markov chain which we call the repeated Markov chain. The repeated Markov chain has only one irreducible class and all its states are positive recurrent. These properties are sufficient to ensure that each repeated-state j has unique equilibrium probability $\Pi_j > 0$, satisfying $\Pi_j = \lim_{n\to\infty} \frac{V_j^n}{n}\ ,\ a.s.$ [21]. ∎

Another characteristic of the repeated-states is that they are either all aperiodic or all periodic (with the same period). The repeated Markov chain is therefore either aperiodic, in which case it is ergodic, or periodic. Regardless of whether the chain is ergodic (aperiodic) or periodic, in order to find the equilibrium probabilities we have to solve the set of equations $\Pi = \Pi P$, $\sum_j \Pi_j = 1$, where P is the transition matrix of the repeated Markov chain, and Π is the equilibrium probability vector.

A.1.2. Repeated-States of the Uniform Distribution

In most cases there is no need to identify the repeated-states in order to perform the analysis. Exceptions are cases where the chain contains more than one irreducible class (in such cases the solution to the set of equations $\Pi = \Pi P$, $\sum_j \Pi_j = 1$ is not unique). However, identifying the repeated-states enables us to reduce the number of states considered in the analysis, resulting in more efficient numerical calculations.

Claim A.1. Let H be any item size 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 under distribution H has a single irreducible class of recurrent states that comprise the repeated-states of the chain. Moreover, the repeated Markov chain in this case is ergodic.

Proof. Denote by S_U the state where all the open bins are full, i.e., $N_t(j_1, j_2, ...j_K) = (U, U, ..U)$. 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 positive recurrent and hence a repeated-state. Since every state leads to S_U, there can be only one subset of recurrent states in the chain; this subset comprise the repeated-states, all other states in the chain are transient.

To show that the repeated-states form an ergodic chain we must now show that the chain is aperiodic. To do so it is sufficient to 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 [21]. ∎

It follows from Claim A.1 that under the $\{u, U\}$ distribution the repeated-states form an ergodic Markov chain for any $u \geq 2$ (the chain is periodic when $u = 1$). We now identify the repeated-states. It is easy to verify that for 1-bounded space algorithms, i.e., NF and SNF, all states communicate and are therefore repeated-states. For $K \geq 2$ this is no longer true. The subset of repeated-states depends on the algorithm.

A.1.3. NF_K Algorithm

Let us first consider the NF_2 algorithm. $N(j_1, j_2)$ is a repeated-state 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 a repeated-state while N'' is transient.

For $K > 2$ a state $N(j_1, j_2, ...j_K)$ is repeated-state 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.1.4. 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 repeated-states of the algorithms are similar to those of NF_K. A state $N(j_1, j_2, ..., j_K)$ is a repeated-state 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 $b = U + 1 - j_{x-1}$.

A.2. EQUILIBRIUM PROBABILITIES AND CONVERGENCE OF LIMITS

In Section 2 we claimed that $\lim_{n\to\infty} E\left[A(L_n)/n\right]$ exists for all the algorithms we consider in this paper, under any discrete item size distribution. We now make this claim concrete. To do so we start by observing that since the overhead accounts for the wasted space in all but the last bin, for any list L_n we have $U \cdot A(L_n) - \sum_{t=1}^{n}(s(a_t) + oh_t) < U$. Now recall that we defined $I_A^n(H) \equiv E\left[\frac{1}{n}\sum_{t=1}^{n}(s(a_t) + oh_t)\right]$ so by taking expectation we get $E[U \cdot A(L_n)] - n \cdot I_A^n(H) < U$. It follows that

$$\lim_{n\to\infty} \left| E\left[\frac{A(L_n)}{n}\right] - \frac{I_A^n(H)}{U} \right| < \lim_{n\to\infty} \frac{1}{n} = 0 \tag{A.2}$$

Now assume that A_K is a K-bounded space algorithm for which the level of the open bins can be described by a finite Markov chain. We have seen that in this case the repeated-states of the Markov chain have equilibrium probabilities under any discrete distribution. When the equilibrium probabilities exist we can use (56) to calculate $I_{A_K}(H)$

$$I_{A_K}(H) = \lim_{n\to\infty} I_A^n(H) = \sum_{j_1=1}^{U} \cdots \sum_{j_K=1}^{U} \sum_{i=1}^{U} h_i \cdot \Pi(j_1, ..., j_K) \cdot (i + oh_i(j_1, ..., j_K)) \tag{A.3}$$

Since the calculation of $I_{A_K}(H)$ does not depend on n we know that $\lim_{n\to\infty} I_A^n(H)$ exists. From (A.2) it now follows that

$$\lim_{n\to\infty} E\left[\frac{A(L_n)}{n}\right] = \lim_{n\to\infty} \frac{I_A^n(H)}{U} = \frac{I_A H}{U} \tag{A.4}$$

We conclude that an algorithm for which the level of the open bins can be described by a finite Markov chain has the property that $\lim_{n\to\infty} E\left[A(L_n)/n\right]$ exists.

To show that $\overline{R}_A^{\infty}(H)$ is well defined note that $OPT(L_n)$ is monotonic increasing (non decreasing) with n; hence, $\lim_{n\to\infty} E\left[OPT(L_n)/n\right]$ exists under any item size distribution H. Since both limits exists we can use (8) to calculate $\overline{R}_A^{\infty}(H)$

$$\overline{R}_A^{\infty}(H) \;=\; \lim_{n\to\infty} E\left[\frac{A(L_n)}{OPT(L_n)}\right] = \lim_{n\to\infty} \frac{E\left[A(L_n)\right]}{E\left[OPT(L_n)\right]} = \frac{I_A(H)}{I_{OPT}(H)} \tag{A.5}$$

Since $s(L_n)$ is also monotonic increasing with n, we can use the same arguments to show that $\overline{\eta}_A^{\infty}(H)$ (defined in (7)) is also well defined.

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:

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

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

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

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 $\Pi_{U-u},...,\Pi_u$

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

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.4) and (B.6) it is relatively easy to find the expected performance ratio when u is close to U. As an example we analyze the cases of $u = U-1$ and $u = U-2$.

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

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

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

The result we obtain agrees with the result reported in [4]. 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 Π_{U-1} in order to compute the expected performance ratio. We find Π_{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 $I_{NF}(\{U-2,U\})$:

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

We can now calculate the expected performance ratio

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

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

As we could see, the calculation of the expected performance ratio is getting more complex as u decreases. To avoid this complexity 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 $\Pi_{U-u}, ..., \Pi_u$. Based on our numerical results we approximate the probabilities $\Pi_{u+1}, ..., \Pi_U$ to be

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

Using the above approximation we can calculate the following sum:

$$\begin{aligned}\sum_{j=u+1}^{U} \Pi_j OH(j) &\approx \sum_{j=u+1}^{U} \frac{3u-U}{2u(u+1)} \frac{(j+u-U)(U-j)}{u} \\ &= \frac{(3u-U)(U-u)(U-u-1)(5u-2U+1)}{12u^2(u+1)}\end{aligned} \tag{B.12}$$

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

$$\overline{R}^{\infty}_{NF}(\{u,U\}) \approx \frac{3U^3-3(5u+1)U^2+u(23u+10)U-3u^2(u+1)}{6u^2(u+1)} \tag{B.13}$$

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 $u < \frac{U}{2}$

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

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

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

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

We use a generating function to find the equilibrium probabilities

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

We obtain the following generating function:

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

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

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

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

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

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

Once we have Π_{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:

$$\overline{R}^{\infty}_{NF}(\{2, U\}) \approx \frac{3U}{3U-1} \tag{B.21}$$

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:

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

To calculate the equilibrium probabilities we must find all u roots of the denominator. We ignore the root $z = 1$ which does not add any information. By substituting each of the remaining $u - 1$ roots in the numerator we get $u - 1$ independent equations. To get an additional equation we use the normalization condition $G(z = 1) = 1$. We now have u linear equation in u unknown, from which the equilibrium probabilities can be calculated.

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

From the above description it is clear that the calculation of the equilibrium probabilities gets harder as u increases. Fortunately, 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

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

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

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

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. When U is sufficiently large ($U > 10$), 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 but remains less than 2%.

APPENDIX C

Obtaining a Closed Form of $I_{SNF}(\{U, U\})$

In subsection 3.2.1 we presented a closed form expression for $I_{SNF}(\{U, U\})$. We show here how this expression is obtained. We present the calculations for the case where U is even, the calculations and the results are slightly different when U is odd. We start with (34) which we repeat here

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

We evaluate each sum in C.1 separately. From the first sum we have

$$\begin{aligned} S_1 \;&=\; \sum_{j=1}^{U/2} \frac{j^2(j-1)}{2U(U-j)(U-j+1)} = \frac{1}{2U} \sum_{j=1}^{U/2} \left(j + 2U + \frac{(3U^2+U)j - 2U^2(U+1)}{(U-j)(U-j+1)} \right) \\ &=\; \frac{1}{2U} \left[\sum_{j=1}^{U/2} (j + 2U) + \sum_{n=U/2}^{U-1} \frac{(3U^2+U)(U-n) - 2U^2(U+1)}{n(n+1)} \right] \\ &=\; \frac{1}{2U} \left[\frac{9U^2+2U}{8} + U \cdot \sum_{n=U/2}^{U-1} \left(\frac{(U^2-U)}{n} - \frac{U^2+2U+1}{n+1} \right) \right] \\ &=\; \frac{1}{2U} \left[\frac{9U^2+2U}{8} + U \cdot \left[(U^2-U)(H_{U-1} - H_{(U-2)/2}) - (U^2+2U+1)(H_U - H_{U/2}) \right] \right] \\ &=\; \frac{17U-6}{16} - \frac{3U+1}{2}(H_U - H_{U/2}) \end{aligned} \tag{C.2}$$

We now evaluate the second sum.

$$\begin{aligned} S_2 &= \sum_{j=U/2+1}^{U} \frac{(U-j)(3j-U-1)}{2Uj} = \frac{1}{2U} \sum_{j=U/2+1}^{U} \frac{-3j^2+(4U+1)j-U(U+1)}{j} \\ &= \frac{7U-2}{16} - \frac{U+1}{2}(H_U - H_{U/2}) \end{aligned} \tag{C.3}$$

Using S_1 and S_2 we obtain the following close form of $I_{SNF}(\{U,U\})$:

$$\begin{aligned} I_{SNF}(\{U,U\}) &= \frac{U+1}{2} + \frac{17U-6}{16} - \frac{3U+1}{2}(H_U - H_{U/2}) \\ &+ \frac{7U-2}{16} - \frac{U+1}{2}(H_U - H_{U/2}) = 2U - (2U+1)(H_U - H_{U/2}) \end{aligned} \tag{C.4}$$

We can get an approximation by using $H_n \approx \ln n$.

$$I_{SNF}(\{U,U\}) \approx 2U(1-\ln 2) - \ln 2 \approx 0.6137U \tag{C.5}$$

When U is odd the expression for $I_{SNF}(\{U,U\})$ is only slightly different

$$I_{SNF}(\{U,U\}) = \frac{2U^2}{U+1} - (2U+1)(H_U - H_{\lceil U/2 \rceil}) \tag{C.6}$$

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

In this section we elaborate on the analysis of the SNF algorithm for the $\{u,U\}$ distribution. Since SNF behaves exactly as NF when $u \le \frac{U}{2}$, the analysis of SNF in this range is identical to the analysis we presented, in Appendix B, for NF. We repeat here only the approximation for the expected performance ratio (see (B.24))

$$\overline{R}^{\infty}_{SNF}(\{u,U\}) \approx \frac{3U}{3U-u+1} \qquad u \le \frac{U}{2} \tag{C.7}$$

When $u > \frac{U}{2}$ SNF is different from NF but the analysis of the two algorithms is similar. From the equation $\Pi = \Pi P$ we derive the following equilibrium equations of SNF:

$$\begin{aligned} \Pi_1 &= \tfrac{1}{u}\Pi_U \\ \Pi_k &= \tfrac{1}{u}\textstyle\sum_{j=1}^{k-1}\Pi_j + \tfrac{1}{u}\sum_{j=U-k+1}^{U}\Pi_j && 2 \le k \le U-u \\ \Pi_k &= \tfrac{1}{U-k}\textstyle\sum_{j=1}^{k-1}\Pi_j + \tfrac{1}{U-k}\sum_{j=U-k+1}^{U}\Pi_j && U-u+1 \le k \le \tfrac{U}{2} \\ \Pi_k &= \tfrac{1}{k} && \tfrac{U}{2}+1 \le k \le u \\ \Pi_k &= \tfrac{1}{u}\textstyle\sum_{j=k-u}^{k-1}\Pi_j && u+1 \le k \le U \end{aligned} \tag{C.8}$$

From (C.8) we get a simple closed form expression for the equilibrium probabilities $\Pi_{U-u}, \ldots, \Pi_u$

$$\Pi_j = \begin{cases} \frac{j}{(U-j)(U-j+1)} & U-u \le j \le \frac{U}{2} \\ \frac{1}{j} & \frac{U}{2} < j \le u \end{cases} \tag{C.9}$$

The equations in (C.8) provide a way of obtaining a closed form expressions for the remaining probabilities also. However, the expressions get quite complex unless u is very close to U. Note that the equilibrium probabilities of states $1 \leq j < U-u$ are not required for the calculation of the expected performance ratio, since the overhead in these states is zero. To calculate the expected performance ratio we must therefore find only the probabilities $\Pi_{u+1}, ..., \Pi_U$. To simplify the solution and to provide more insight, we use an approximation for these probabilities. The approximation is based on analysis of several values of u and on our numerical calculations.

$$\Pi_j \approx \frac{j}{uU} \quad u < j \leq U \tag{C.10}$$

The average overhead of SNF was given in (39) from which we obtain the following expression for $OH(j)$:

$$OH(j) = \begin{cases} \frac{j(j-1)-(U-u)(U-u-1)}{2u} & U-u+1 \leq j \leq \frac{U}{2} \\ \frac{2(2j-U)(U-j)+2(u-j)U-u(u+1)+j(j+1)}{2u} & \frac{U}{2} < j \leq u \\ \frac{(u-U+j)(U-j)}{u} & u < j \leq U \end{cases} \tag{C.11}$$

We use (C.9) (C.10) and (C.11) to calculate an approximation of $I_{SNF}(\{u, U\})$

$$\begin{aligned} I_{SNF}(\{u,U\}) &= \frac{u+1}{2} + \sum_{j=U-u+1}^{U} \Pi_j \cdot OH(j) \approx \frac{(u^2(u+1)(7u+5) - 2u(3+4u(4u+3))U}{12Uu^2} \\ &+ \frac{(1+6u(13u+2))U^2 - (28u+U)U^3)}{12Uu^2} - \frac{(2U+1)U(H_u - H_{U/2})}{u} \end{aligned} \tag{C.12}$$

We can now calculate the asymptotic expected performance ratio

$$\overline{R}^{\infty}_{SNF}(\{u,U\}) = \frac{2I_{SNF}(\{u,U\})}{u+1} \tag{C.13}$$

When U is sufficiently large ($U > 10$), the approximation error is negligible when $u \approx U$. As u becomes closer to $\frac{U}{2}$ the approximation error increases. For example, when $U = 100$ and $u = 70$ the approximation error is less than 1% but when $u = 50$ the approximation error is 5%.

APPENDIX D

Computing Numerical Results

In this section we briefly describe some of the technical details we used in calculating the numerical results presented in this paper.

In our analysis we first construct a transition matrix P from which we calculate the equilibrium probabilities. When analyzing a K-bounded space algorithm there are U^K possible states which means we must construct a $U^K \times U^K$ transition matrix. The complexity of a naive implementation is therefore $O(U^{2K})$, in both time and memory. There are several properties of the analysis we can exploit in order to reduce the complexity of the problem. First note that we can reduce the number of states by removing all transient states, as there is no need to include them in our asymptotic analysis. The number of transient states depends on the algorithm and on the item size distribution (see Appendix A for details). In addition in some cases we may reduce the number of states by using the properties of the algorithm. For example, our state definition of BBF_2 reduced the number of states from U^2 to $\frac{U(U+1)}{2}$. Another important factor is the number of non zero elements in P. Note that if there are s item sizes, each line in P contains at most s non zero elements. The matrix P is therefore very sparse with at most $s \cdot U^K$ non zero elements.

We found the MATLAB© software a very convenient tool for performing our calculations. MATLAB provides powerful functions for manipulating matrices and includes built in support for representing and handling sparse matrices. Reducing the number of states and using a sparse representation enabled us to overcome the huge memory requirements involved in calculating results for high values of U. Our MATLAB programs (scripts) are very simple and require about 30 lines of code.

We used a standard Pentium© III PC with 128MB for our calculations. Calculating results for 1-bounded space algorithms (for cases where we have no closed form) was straightforward and presented no problem. For example, calculating $\overline{R}^{\infty}_{NF}(\{u,U\})$ even for $U = 1000$ required less than 30 seconds. To calculate results for 2-bounded space algorithms we used several optimizations. Our optimizations included reducing the number of states, using a sparse representation of the transition matrix, and using an iterative method for calculating the equilibrium probabilities. Calculating results for $U < 100$ took only several minutes. However, in order to calculate results for $U = 300$ we had to run the program for more than a week; in this case constructing the matrix took more than 98% of the time.

We believe our calculations can be improved but we have made no attempt to optimize them any further, as this was not the goal of our research. It is clear that there is a limit to our ability to calculate results in this way since the time and memory complexities of the calculations are at least $O(U^K)$. This means that calculating the performance ratio for higher values of K can only be done for very small values of U.

APPENDIX E

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.

Recall that the expected performance ratio of algorithm A under item-size distribution H is defined as

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

The asymptotic expected performance ratio is defined as

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

Let A be a bin packing algorithm and let R_A be the worst case performance ratio of A. We show that if $R_A = O(1)$ the ratio of expectations converges to the same limit of $E[A(L_n)/OPT(L_n)]$ when $n \to \infty$. For convenience we adopt the standard bin packing convention and assume that item sizes are in the range $(0, 1]$.

Claim E.1. For any item-size distribution H and any algorithm A with $R_A = O(1)$

$$\lim_{n\to\infty} \left| E\left[\frac{A(L_n)}{OPT(L_n)}\right] - \frac{E[A(L_n)]}{E[OPT(L_n)]} \right| = 0 \tag{E.3}$$

Proof: To prove the claim we use the fact that the optimal packing has the following property under any item size distribution [24]:

$$Pr\left(|\,OPT(L_n) - E[OPT(L_n)]\,| \geq t\right) \leq 2e^{\frac{-t^2}{2n}} \tag{E.4}$$

We use the above property and the fact that $A(L_n) \leq C \cdot OPT(L_n)$, where C is a constant, to prove the claim.

$$\begin{aligned}
\lim_{n\to\infty} & \left| E\left[\frac{A(L_n)}{OPT(L_n)}\right] - \frac{E[A(L_n)]}{E[OPT(L_n)]} \right| \\
&= \lim_{n\to\infty} \left| E\left[\frac{A(L_n)}{OPT(L_n)}\right] - E\left[\frac{A(L_n)}{E[OPT(L_n)]}\right] \right| \\
&= \lim_{n\to\infty} \left| E\left[\frac{A(L_n)}{OPT(L_n)}\left[1 - \frac{OPT(L_n)}{E[OPT(L_n)]}\right]\right] \right| \\
&\leq \lim_{n\to\infty} E\left[\left| \frac{A(L_n)}{OPT(L_n)}\left[1 - \frac{OPT(L_n)}{E[OPT(L_n)]}\right] \right|\right] \\
&= \lim_{n\to\infty} E\left[\left| \frac{A(L_n)}{OPT(L_n)} \right| \cdot \left| \frac{E[OPT(L_n)] - OPT(L_n)}{E[OPT(L_n)]} \right|\right] \\
&\leq \lim_{n\to\infty} R_A \cdot \frac{1}{n\overline{h}} \cdot E\left[|\,E[OPT(L_n)] - OPT(L_n)\,|\right]
\end{aligned}$$

Where $\overline{h}$ 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.

$$\begin{aligned}
&\lim_{n\to\infty} R_A^n \cdot \frac{1}{n\overline{h}} \cdot E\left[|\,E[OPT(L_n)] - OPT(L_n)\,|\right] \\
&\leq \lim_{n\to\infty} R_A^n \cdot \frac{1}{n\overline{h}} \int_0^\infty Pr\left(|\,E[OPT(L_n)] - OPT(L_n)\,| \geq t\right) dt \\
&\leq \lim_{n\to\infty} R_A^n \cdot \frac{1}{n\overline{h}} \int_0^\infty 2e^{\frac{-t^2}{2n}}\, dt = \lim_{n\to\infty} R_A^n \cdot \frac{1}{n\overline{h}} \sqrt{2n\pi}
\end{aligned}$$

We conclude that the claim holds for any item-size distribution H and any algorithm A whose worst case performance ratio satisfies $R_A = o(\sqrt{n})$. In such cases we have

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

∎