Computing 50, 265–270 (1993)

Computing

Short Communication/Kurze Mitteilung
Best-*k*-Fit Bin Packing

W. Mao, Williamsburg, VA

Received March 26, 1992; revised October 9, 1992

Abstract — Zusammenfassung

Best-*k*-Fit Bin Packing. The bin packing problem is to pack a list of reals in (0, 1] into unit-capacity bins using the minimum number of bins. Let $R^\infty[A]$ be the limiting worst value for the ratio $A(L)/OPT(L)$ as $OPT(L)$ goes to ∞, where $A(L)$ denotes the number of bins used when the approximation algorithm A is applied to the list L, and $OPT(L)$ denotes the minimum number of bins needed to pack L. For Best-k-Fit (BkF for short, $k \geq 1$), we prove in this paper that $R^\infty[BkF] = 1.7 + \frac{3}{10k}$.

AMS Subject Classifications: 68Q25, 68R05

Key words: Bin packing, approximation algorithms, best-k-fit algorithm, worst-case performance bound.

Best-*k*-Fit Bin Packing. Das Bin-Packing-Problem besteht darin, eine Liste von reellen Zahlen aus (0, 1], in Kästen ("bins") der Kapazität 1 zu verpacken. Sei $A(L)$ die Zahl der Kästen, die benötigt wird, wenn der Approximationsalgorithmus A auf die Liste L angewandt wird und $OPT(L)$ die Minimalzahl der für das Packen von L benötigten Kästen. Sei weiter $R^\infty[A]$ der Grenzwert des Verhältnisses $A(L)/OPT(L)$ für OPT $(L) \to \infty$ im ungünstigsten Fall. Für Best-k-Fit (kurz BkF, $k \geq 1$) zeigen wir in diesem Beitrag, daß $R^\infty[BkF] = 1.7 + \frac{3}{10k}$.

1. Introduction

In the bin packing problem, we are given a list $L = (a_1, a_2, \ldots, a_n)$, where each number a_i is in (0, 1], and are asked to find a packing of these numbers into a minimum number of unit-capacity bins. Since the problem of finding an optimal packing for any list L is NP-complete, research has focused on finding near-optimal approximation algorithms.

Definition: For any approximation algorithm A and any list L, let $A(L)$ be the number of bins used when A is applied to L, and $OPT(L)$ be the minimum number of bins needed to pack L. $R_n[A]$ is defined to be $\max\{A(L)/OPT(L): |L| = n\}$, and the *worst-case performance bound* of A is defined to be $R^\infty[A] = \limsup_{n\to\infty} R_n[A]$.

In [2], Csirik and Johnson give the following definition of what they call *k-bounded space* on-line bin packing algorithm.

Definition: An on-line bin packing algorithm uses *k-bounded space* if for each number a_i, the choice for the bin to place it is restricted to a set of k or fewer *active* bins, where each bin becomes active when it gets its first number, but once it is declared *closed*, it can never be active again.

Which bin among the k active bins to choose for packing a_i, and which bin among the k active bins to choose for closing when a new bin has to be created for a_i? In the most straightforward fashion, we always choose the first bin, i.e., the lowest indexed or the oldest, among the k active bins into which a_i can fit, and close the first bin among the k bins when a new bin has to be created. This algorithm is called Next-k-Fit (NkF). It is proved that $R^\infty[N1F] = 2$, and $R^\infty[NkF] = 1.7 + \frac{3}{10(k-1)}$ for $k \geq 2$ [1] [4] [5]. However, in the other extremal case, we can choose the fullest bin (with the maximum current capacity) among the k active bins into which a_i can fit, and close the fullest bin among the k bins when a new bin has to be created. This algorithm is called k-Bounded Best Fit (BBF_k). It is proved that $R^\infty[BBK_1] = 2$, and $R^\infty[BBF_k] = 1.7$ for $k \geq 2$ [2].

In this paper, we study the algorithm, in which the fullest among the k active bins is chosen for a_i, and lowest-indexed active bin is closed when a new bin has to be created. We call this algorithm Best-k-Fit (BkF). Intuition tells us that the performance of BkF is better than NkF, but worse than BBF_k. It is clear that $R^\infty[B1F] = R^\infty[N1F] = R^\infty[BBF_1] = 2$. For $k \geq 2$, our result is that $R^\infty[BkF] = 1.7 + \frac{3}{10k}$.

In §2, we prove the lower bound of $R^\infty[BkF]$. In §3, we prove the upper bound of $R^\infty[BkF]$ by using the traditional weighting function method and a case study similar to but much simpler than that in the NkF proof [5]. Finally, §4 gives the conclusion.

2. Lower Bound of $R^\infty[BkF]$

Usually, a lower bound result can be obtained by creating a special list L, generating its optimal packing and BkF packing, and computing its performance ratio $BkF(L)/OPT(L)$, whose limiting value is actually a lower bound of $R^\infty[BkF]$.

Theorem 1: *For $k \geq 2$, $R^\infty[BkF] \geq 1.7 + \frac{3}{10k}$.*

Proof: The list given by Johnson et al. in the proof of $R^\infty[NkF] \geq 1.7 + \frac{3}{10k}$ [3] yields the same lower bound for $R^\infty[BkF]$. □

3. Upper Bound of $R^\infty[BkF]$

3.1 Upper Bound Proving Technique

Define a weighting function W as follows.

$$W(\alpha) = \begin{cases} \frac{6}{5}\alpha & \text{if } \alpha \in (0, \frac{1}{6}] \\ \frac{9}{5}\alpha - \frac{1}{10} & \text{if } \alpha \in (\frac{1}{6}, \frac{1}{3}] \\ \frac{6}{5}\alpha + \frac{1}{10} & \text{if } \alpha \in (\frac{1}{3}, \frac{1}{2}] \\ \frac{6}{5}\alpha + \frac{2}{5} + \frac{3}{10k} & \text{if } \alpha \in (\frac{1}{2}, 1] \end{cases}$$

With the weighting function, we define the weight of a number a in list L to be $W(a)$, the weight of a bin B in a packing to be $W(B) = \sum_{a \in B} W(a)$, and the weight of a list L to be $W(L) = \sum_{a \in L} W(a)$.

In §3.2 and §3.3, we will prove that $BkF(L) \leq W(L) + c$ and $W(L) \leq (1.7 + \frac{3}{10k})OPT(L)$ for any L, which together imply the $1.7 + \frac{3}{10k}$ upper bound of $R^{\infty}[BkF]$.

3.2 A Lemma

Lemma: *Let $B_1, B_2, \ldots$ be the BkF packing of any infinitely long list L. If $B_i < \frac{5}{6}$ for some i, then there always exists $j > 0$ such that $\frac{6}{5}B_i + W(B_{i+1}) + \cdots + W(B_{i+j-1}) + W(B_{i+j}) \geq j + \frac{6}{5}B_{i+j}$.*

Proof: If the sum of the numbers in B_i, also denoted by B_i without confusion, is no greater than $\frac{1}{2}$, then bin B_i must be followed by k bins with their bottommost numbers greater than $\frac{1}{2}$. The lemma holds with $j = k$ because $\frac{6}{5}B_i + W(B_{i+1}) + \cdots + W(B_{i+k}) \geq \frac{6}{5}(B_i + B_{i+1}) + \frac{2}{5} + \frac{3}{10k} + (\frac{6}{5} \times \frac{1}{2} + \frac{2}{5} + \frac{3}{10k})(k-2) + \frac{6}{5}B_{i+k} + \frac{2}{5} + \frac{3}{10k} \geq \frac{6}{5} \times 1 + k - \frac{9}{10} + \frac{6}{5}B_{i+k} \geq k + \frac{6}{5}B_{i+k}$.

If B_{i+1} has one number greater than $\frac{1}{2}$, the lemma holds with $j = 1$ because $\frac{6}{5}B_i + W(B_{i+1}) \geq \frac{6}{5} \times \frac{1}{2} + \frac{6}{5}B_{i+1} + \frac{2}{5} + \frac{3}{10k} \geq 1 + \frac{6}{5}B_{i+1}$.

From now on, we assume that $\frac{1}{2} < B_i < \frac{5}{6}$ and that all numbers in B_{i+1} are no greater than $\frac{1}{2}$. We introduce notation $A_{s,t}$ for any bin B_s, where $1 \leq t \leq k$, to be the sum of the numbers coming to B_s when B_s is the t^{th} youngest among the k active bins in the BkF packing (Fig. 1). Clearly, $A_{s,1} + \cdots + A_{s,k} = B_s$.

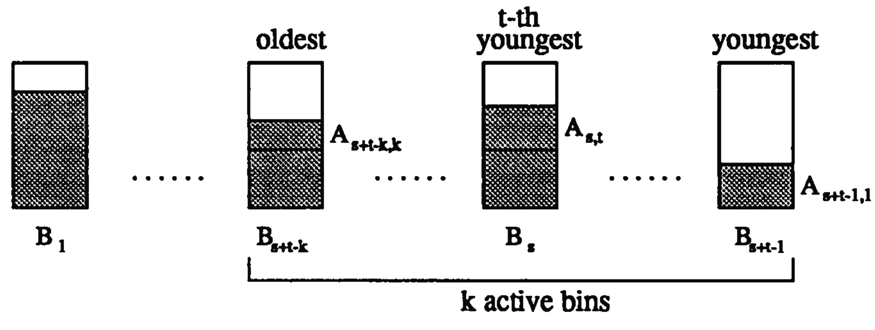

Figure 1. The formation of $A_{s,t}$

Case 1: $A_{i+1,1} + \cdots + A_{i+1,k-1} \leq \frac{1}{2}$.

If $A_{i+1,1} + \cdots + A_{i+1,k-1} + A_{i+1,k} \leq \frac{1}{2}$, then B_{i+1} is followed by k bins with their bottommost numbers greater than $\frac{1}{2}$. Since in B_{i+1} the bottommost number $b_{i+1} > 1 - B_i$, and $\frac{1}{6} < b_{i+1} \leq \frac{1}{2}$, then $\frac{6}{5}B_i + W(b_{i+1}) - \frac{6}{5}b_{i+1} \geq \frac{6}{5}B_i + \min\{\frac{3}{5}(1 - B_i) - \frac{1}{10}, \frac{1}{10}\} \geq \min\{\frac{3}{5} \times \frac{1}{2} + \frac{5}{10}, \frac{6}{5} \times \frac{1}{2} + \frac{1}{10}\} \geq \frac{7}{10}$. Therefore, $\frac{6}{5}B_i + W(B_{i+1}) + \cdots +$

$W(B_{i+k+1}) \geq \frac{6}{5}B_i + W(b_{i+1}) - \frac{6}{5}b_{i+1} + \frac{6}{5}(B_{i+1} + B_{i+2}) + \frac{6}{5} \times \frac{1}{2}(k-2) + (\frac{2}{5} + \frac{3}{10k})k + \frac{6}{5}B_{i+k+1} \geq \frac{7}{10} + \frac{6}{5} \times 1 + k - \frac{9}{10} + \frac{6}{5}B_{i+k+1} \geq k + 1 + \frac{6}{5}B_{i+k+1}$. So the lemma holds with $j = k + 1$.

If $A_{i+1,1} + \cdots + A_{i+1,k-1} + A_{i+1,k} > \frac{1}{2}$, then B_{i+1} is followed by $k-1$ bins with their bottommost numbers greater than $\frac{1}{2}$, and all the numbers in $A_{i+1,1}, \ldots, A_{i+1,k-1}$ are greater than $\frac{1}{6}$ (otherwise they would go to the fuller B_i in *BkF*). Let a be the smallest number in $A_{i+1,k}$ such that right before a is packed into B_{i+1}, the content of B_{i+1} is no greater than $\frac{1}{2}$. Clearly, $a + B_{i+2} > 1$ because of the nature of *BkF*. We claim that $W(B_{i+1}) \geq \frac{6}{5}(b_{i+1} + a) + \frac{1}{20}$. If B_{i+1} has at least one number no less than $\frac{1}{4}$, then $W(B_{i+1}) \geq \frac{6}{5}(b_{i+1} + a) + \min\{\frac{3}{5} \times \frac{1}{4} - \frac{1}{10}, \frac{1}{10}\} \geq \frac{6}{5}(b_{i+1} + a) + \frac{1}{20}$. Otherwise, if all the numbers in B_{i+1} are less than $\frac{1}{4}$, there are at least three numbers in B_{i+1} since $B_{i+1} > \frac{1}{2}$. If $A_{i+1,1}, \ldots, A_{i+1,k-1}$ together have more than one number, then $W(B_{i+1}) \geq \frac{6}{5}(b_{i+1} + a) + \frac{6}{5} \times \frac{1}{6} \geq \frac{6}{5}(b_{i+1} + a) + \frac{1}{20}$. If $A_{i+1,1}, \ldots, A_{i+1,k-1}$ have only one number b_{i+1}, which is less than $\frac{1}{4}$, then $A_{i+1,k} > \frac{1}{4}$ and it contains a number d other than a such that right before it is packed into B_{i+1} the content of B_{i+1} is no greater than $\frac{1}{2}$. So $a \leq d$, and $A_{i+1,k} - a > \frac{1}{2} \times \frac{1}{4}$. Then $W(B_{i+1}) \geq \frac{6}{5}(b_{i+1} + a) + \frac{6}{5}(A_{i+1,k} - a) \geq \frac{6}{5}(b_{i+1} + a) + \frac{1}{20}$. Therefore, $\frac{6}{5}B_i + W(B_{i+1}) + \cdots + W(B_{i+k+1}) \geq \frac{6}{5}B_i + \frac{6}{5}(b_{i+1} + a) + \frac{1}{20} + \frac{6}{5}B_{i+2} + \frac{6}{5} \times \frac{1}{2}(k-2) + (\frac{2}{5} + \frac{3}{10k})(k-1) + \frac{6}{5}B_{i+k+1} \geq \frac{6}{5}(B_i + b_{i+1}) + \frac{6}{5}(a + B_{i+2}) + k - \frac{7}{5} + (\frac{3}{20} - \frac{3}{10k}) + \frac{6}{5}B_{i+k+1} \geq \frac{6}{5} \times 1 + \frac{6}{5} \times 1 + k - \frac{7}{5} + \frac{6}{5}B_{i+k+1} \geq k + 1 + \frac{6}{5}B_{i+k+1}$. So the lemma holds with $j = k + 1$.

Case 2: $A_{i+1,1} + \cdots + A_{i+1,k-1} > \frac{1}{2}$. In $A_{i+1,1}, \ldots, A_{i+1,k-1}$, there are at least two numbers since no number is greater than $\frac{1}{2}$, and the two bottommost numbers are greater than $\frac{1}{6}$ since otherwise they would go to the fuller B_i. In this case, we assume that at least one of the two bottommost numbers in $A_{i+1}, \ldots, A_{i+1,k-1}$ is in $(\frac{1}{6}, \frac{1}{3}]$. The other one must be either in $(\frac{1}{6}, \frac{1}{3}]$ or $(\frac{1}{3}, \frac{1}{2}]$. It is clear that $B_i > \frac{2}{3}$. Therefore, $\frac{6}{5}B_i + W(B_{i+1}) \geq \frac{6}{5}B_i + \frac{6}{5}B_{i+1} + \frac{3}{5}(1 - B_i) - \frac{1}{10} + \min\{\frac{3}{5}(1 - B_i) - \frac{1}{10}, \frac{1}{10}\} \geq \frac{6}{5}B_{i+1} + \min\{1, \frac{3}{5}B_i + \frac{3}{5}\} \geq 1 + \frac{6}{5}B_{i+1}$. So the lemma holds with $j = 1$.

Case 3: $A_{i+1,1} + \cdots + A_{i+1,k-1} > \frac{1}{2}$ with the two bottommost numbers both in $(\frac{1}{3}, \frac{1}{2}]$, and $A_{i+2,1} + \cdots + A_{i+2,k-1} \leq \frac{1}{2}$. When $B_i \geq \frac{2}{3}$, the lemma holds with $j = 1$ because $\frac{6}{5}B_i + W(B_{i+1}) \geq \frac{6}{5} \times \frac{2}{3} + \frac{6}{5}B_{i+1} + \frac{1}{10} + \frac{1}{10} \geq 1 + \frac{6}{5}B_{i+1}$. When $B_i < \frac{2}{3}$ and $B_{i+1} \geq \frac{6}{5}$, the lemma holds with $j = 2$ because $\frac{6}{5}B_i + W(B_{i+1}) + W(B_{i+2}) \geq \frac{6}{5} \times 1 + \frac{6}{5} \times \frac{5}{12} + \frac{1}{10} + \frac{1}{10} + \frac{6}{5}B_{i+2} + \frac{1}{10} \geq 2 + \frac{6}{5}B_{i+2}$. When B_{i+2} has a number greater than $\frac{1}{2}$, the lemma holds with $j = 2$ because $\frac{6}{5}B_i + W(B_{i+1}) + W(B_{i+2}) \geq \frac{6}{5} \times \frac{1}{2} + \frac{6}{5} \times \frac{2}{3} + \frac{1}{10} + \frac{1}{10} + \frac{6}{5}B_{i+2} + \frac{2}{5} + \frac{3}{10k} \geq 2 + \frac{6}{5}B_{i+2}$. Let us assume $B_i < \frac{2}{3}$, $B_{i+1} < \frac{5}{6}$ and all numbers in B_{i+2} are no greater than $\frac{1}{2}$. Comparing this case with case 1, we find that the only difference is that in this case there is a bin B_{i+1} with $W(B_{i+1}) \geq 1$ between B_i and $B_{i+2}, B_{i+3}, \ldots, B_{i+k+2}$. Using the exactly same method as that in case 1, we can prove that $\frac{6}{5}B_i + W(B_{i+2}) + \cdots + W(B_{i+k+2}) \geq k + 1 + \frac{6}{5}B_{i+k+2}$. Therefore, $\frac{6}{5}B_i + W(B_{i+1}) + W(B_{i+2}) + \cdots + W(B_{i+k+2}) \geq k + 2 + \frac{6}{5}B_{i+k+2}$. So the lemma holds with $j = k + 2$.

Case 4: $A_{i+1,1} + \cdots + A_{i+1,k-1} > \frac{1}{2}$ with the two bottommost numbers both in $(\frac{1}{3}, \frac{1}{2}]$, and $A_{i+2,1} + \cdots + A_{i+2,k-1} > \frac{1}{2}$. For the same reasons as in case 3, the lemma holds with $j = 1$ when $B_i \geq \frac{2}{3}$, $j = 2$ when $B_i < \frac{2}{3}$ and $B_{i+1} \geq \frac{5}{6}$, and $j = 2$ when B_{i+2} has a number greater than $\frac{1}{2}$. Let us assume $B_i < \frac{2}{3}$, $B_{i+1} < \frac{5}{6}$ and all numbers in B_{i+2}

are no greater than $\frac{1}{2}$. If the two bottommost numbers in B_{i+2} are both in $(\frac{1}{3}, \frac{1}{2}]$, then $\frac{6}{5}B_i + W(B_{i+1}) + W(B_{i+2}) \geq \frac{6}{5} \times (1 + \frac{1}{3}) + \frac{1}{10} \times 4 + \frac{6}{5}B_{i+2} \geq 2 + \frac{6}{5}B_{i+2}$. So the lemma holds with $j = 2$. If one of the two bottommost numbers in B_{i+2} is in $(\frac{1}{6}, \frac{1}{3}]$, then $\frac{6}{5}B_i + W(B_{i+1}) + W(B_{i+2}) \geq \frac{6}{5}B_i + \frac{6}{5}B_{i+1} + \frac{1}{10} + \frac{1}{10} + \frac{6}{5}B_{i+2} + \frac{1}{10} + \frac{3}{5}(1 - B_{i+1}) - \frac{1}{10} \geq \frac{6}{5}B_i + \frac{3}{5}(1 - B_i + 1 - B_i) + \frac{4}{5} + \frac{6}{5}B_{i+2} \geq 2 + \frac{6}{5}B_{i+2}$. So the lemma holds with $j = 2$.

This completes the proof of the lemma. □

3.3 *The Proof of the Upper Bound*

Claim 1: *For any list L, $W(L) \leq (1.7 + \frac{3}{10k})OPT(L)$.*

Proof: By a similar case analysis as in the proof of Claim 1 in [5], we can show that $W(B) \leq 1.7 + \frac{3}{10k}$. For any list L, we apply the optimal algorithm. $W(L) = \sum_{i=1}^{OPT(L)} W(B_i) \leq \sum_{i=1}^{OPT(L)} (1.7 + \frac{3}{10k}) = (1.7 + \frac{3}{10k})OPT(L)$. □

Claim 2: *For any list L, there is $c \geq 0$ such that $BkF(L) \leq W(L) + c$.*

Proof: For any list L, we get $BkF(L)$ nonempty bins after BkF is applied to the list. For each bin, the weight of the bin is computed. The following pseudo-pascal code describes the checking procedure:

```
i := 1;
repeat
  if W(B_i) ≥ 1 then i := i + 1
  else if B_i, B_{i+1}, ... matches any of the cases discussed above
    then begin
         By the lemma there exists j such that 6/5 B_i + W(B_{i+1}) + ··· +
         W(B_{i+j}) − 6/5 B_{i+j} ≥ j, i.e., the average weight of each bin among
         B_i, ..., B_{i+j−1} is greater than or equal to 1 after borrowing weight
         W(B_{i+j}) − 6/5 B_{i+j} from B_{i+j};
         W(B_{i+j}) := 6/5 B_{i+j};
         i := i + j
         end
    else begin /*The packing ends without a complete maching.*/
         The constant c is used to make up the weight shortfall so that the
         average weight of each bin among B_i, ..., B_{BkF(L)} is larger than 1;
         i := BkF(L)
         end
until i = BkF(L);
```

Following this procedure, we know there is always a way to distribute the total weight $W(L)$ such that each bin has an average weight of 1. Therefore, $W(L) + c \geq BkF(L)$. □

Theorem 2: *For $k \geq 2$, $R^{\infty}[BkF] \leq 1.7 + \frac{3}{10k}$.*

Proof: The proof is straightforward using Claim 1 and Claim 2. □

The worst-case performance bound of BkF is $1.7 + \frac{3}{10k}$ since both the lower bound and upper bound of $R^\infty[BkF]$ are $1.7 + \frac{3}{10k}$.

4. Conclusion

As indicated by Csirik and Johnson [2], in the k-bounded space on-line algorithms, choosing the fullest bin for a_i results in better performance than choosing the first bin, and closing the fullest bin results in better performance than closing the first bin. Compared with the other two well-studied k-bounded space algorithms, BkF performs better than NkF but worse than BBF_k in the worst case, which further justifies the statement made by Csirik and Johnson that best is better than first.

Acknowledgements

I wish to thank Paul Stockmeyer for bringing the Best-k-Fit algorithm to my attention, Jianwei Zhang and Chris Michael for helping me translate the abstract into German, and the referee for finding out a mistake in Case 4 of the case analysis.

References

[1] Csirik, J., Imreh, B.: On the worst-case performance of the NkF bin-packing heuristic. Acta Cybernetica *9*, 89–105 (1989).
[2] Csirik, J., Johnson, D. S.: Bounded space on-line bin packing: Best is better than first. Proc. 2nd Ann. ACM-SIAM Symp. on Discrete Algorithms, pp. 309–319 (1991).
[3] Johnson, D. S.: Fast algorithms for bin packing. J. Comput. System Sci. *8*, 274–314 (1974).
[4] Mao, W.: Scheduling and bin packing: A study of the worst-case performance bounds. Ph.D thesis, Department of Computer Science, Princeton University (1990).
[5] Mao, W.: Tight worst-case performance bounds for Next-k-Fit bin packing. SIAM J. on Comput. *22*, 46–56 (1993).

Weizhen Mao
Department of Computer Science
College of William and Mary
Williamsburg, VA 23185
U.S.A.