We can assume that

$$Area(|R_{i+1} - R_i|) = p^2\sqrt{t_i}/2^i \geq p^2 2^{-i}\sqrt{\tfrac{1}{\log n}}.$$

We can make this assumption since by Lemma 3.2.20 increasing the area increases the probability of a large discrepancy. We then apply Lemma 3.2.19. This lemma says that for a particular value of t_i and a particular $R_{i+1} - R_i$, the discrepancy is at most $\sigma\sqrt{A}$ with probability $1 - e^{-\sigma^2/3}$, provided $\sigma^2 \leq A/2$. The bound for $\Delta(R_{i+1} - R_i)$ above is $\sigma\sqrt{A}$ when

$$\sigma = \frac{3pt_i^{1/4}\sqrt{\log(t_i \log n)}}{\sqrt{p^2\sqrt{t_i}\;2^{-i}}} = 3 \cdot 2^{i/2}\sqrt{\log(t_i \log n)}.$$

We now check that $\sigma^2 \leq A/2$. Squaring the expression we have for σ, we get

$$\sigma^2 = 9 \cdot 2^i \log(t_i \log n) \leq 9 \cdot 2^i \log(C \log n).$$

Thus, it is sufficient to prove that

$$2^i(\log C \log n) = o(p^2 2^{-i} \log^{-1/2} n), \quad \text{or} \quad 2^{2i}\sqrt{\log n}\log\log n = o(p^2).$$

Recall that $2^i \leq 2^m = \Theta(p/\log^{3/4} n)$. Substituting this in the above equation, we obtain

$$2^{2i}\sqrt{\log n}\log\log n = O(p^2 \log\log n/\log n) = o(p^2),$$

so we do have $\sigma^2 \leq A/2$, and we can apply Lemma 3.2.19.

Thus, for a given t_i and a given $R_{i+1} - R_i$, the discrepancy is larger than $3pt_i^{1/4}\sqrt{\log(t_i \log n)}$ with probability $e^{-3 \cdot 2^i \log(t_i \log n)}$. Since there are only $2^{2^{i+1}\log(t_i \log n)}$ regions $R_{i+1} - R_i$ with a given t_i, it is larger than desired for fixed values of t_i with probability at most

$$\left(4e^{-3}\right)^{2^i \log(t_i \log n)}.$$

Since $i \geq 2 \log \log n$, $\log(t_i \log n) \geq 1$, and $4e^{-3} \leq\leq \frac{1}{2}$, this expression is less than $n^{-\log n}$. There are only $O(\log n)$ possible values of i and only $O(\log \log n)$ possible values of t_i, so we have that with probability $1 - n^{-\log^{1/2} n}$, that for every $R_{i+1} - R_i$ with $i \geq 2 \log \log n$,

$$\Delta(R_{i+1} - R_i) \leq 3pt_i^{1/4} \sqrt{\log(t_i \log n)}.$$

We must now show that if $\sum_i t_i \leq C$, then

$$\sum_i 3pt_i^{1/4} \sqrt{\log(t_i \log n)} = O(p \log^{3/4} n.)$$

We do this by finding the maximum of

$$\sum_{i=1}^{\log n} t_i^{1/4} \sqrt{\log(t_i \log n)}$$

given that $\sum_i t_i \leq C$. By Jensen's inequality the maximum occurs when all the values of t_i are equal because $t_i^{1/4} \sqrt{\log(t_i \log n)}$ is a convex function. This can be proven by showing that the second derivative of the function is negative. It is more easily done, though, by showing that this function is the geometric mean of two convex functions: $t_i^{1/2}$ and $\log(t_i \log n)$.

We must now bound the total discrepancy of the first $2 \log \log n$ levels.

Theorem 3.2.22: The total sum of the discrepancies of the regions $R_{i+1} - R_i$, for $i \leq 2 \log \log n$, is with probability $1 - n^{-(\log n)^{1/2-\epsilon}}$ at most $O(p \log^{3/4} n)$ for any $\epsilon > 0$.

Proof: We show that, even in the worst case (where t_i is as large as it can get, maximizing the area of and the number of possibilities for $R_{i+1} - R_i$, the discrepancy of the first $2 \log \log n$ levels is $O(p \log^{3/4} n)$ with high probability. As before, increasing the area only increases the chance of a large discrepancy. Thus, we can assume that the area of $|R_{i+1} - R_i|$ is $c_1 p^2 / 2^i$ for some constant c_1 since

$$Area(R_{i+1} - R_i) \leq \frac{p^2 \sqrt{t_i}}{2^i} \leq \frac{p^2 \sqrt{C}}{2^i}.$$

The number of possibilities for $R_{i+1} - R_i$ is

$$n^3 2^{2 \cdot 2^i \log(C \log n)} \leq 2^{3 \cdot 2^i \log\log n + 3 \log n}$$

for sufficiently large n.

Since there are $2^{3 \cdot 2^i \log\log n + 3 \log n}$ possible regions $R_{i+1} - R_i$, if we want something to be true for all of them with probability $1 - 2^{-(\log n)^{3/2-\epsilon}}$ it must be true for any one of them with probability

$$q = 1 - 2^{-3 \cdot 2^i \log\log n - (\log n)^{3/2-\epsilon}}.$$

Assuming the hypotheses of Lemma 3.2.19 hold, this lemma says that with probability q

$$\Delta(R_{i+1} - R_i)^2 \leq 3c_1 p^2 2^{-i} \left(2^i \log\log n + (\log n)^{3/2-\epsilon}\right).$$

This is sufficient to prove the result, since this implies that with high probability the discrepancy is at most

$$O\left(p\sqrt{\log\log n} + p2^{-i/2}(\log n)^{3/4-\epsilon/2}\right).$$

Summing this over the first $\log\log n$ levels, we find that the first term sums to $O(p(\log\log n)^{3/2})$ and that the second is a geometric series which sums to $O(p(\log n)^{3/4-\epsilon/2})$. Thus, the total discrepancy is $O(p(\log n)^{3/4-\epsilon/2})$, as desired.

We must still check the hypotheses of Lemma 3.2.19. That is, we must check that

$$\frac{p^2}{2^{i+1}} \geq 3 \cdot 2^i \log\log n + (\log n)^{3/2-\epsilon}$$

First we show that

$$2^{2i} \log\log n = o(p^2).$$

This holds since $2^i \leq p/\log^{3/4} n$, so $2^{2i} \leq p^2/\log^{3/2} n$. We must also show that

$$(\log n)^{3/2-\epsilon} = o(p^2/2^i).$$

This follows from the inequalities $p/2^i \geq \log^{3/4} n$ and $p \geq \log^{3/4} n$. This proves Theorem 3.2.22, completing the proof of the upper bound for maximum edge length matching. ▮

Part II. Bin Packing

Chapter 4. Introduction

In Part I of this thesis, we investigated several planar matching problems. In Part II, we will use these problems to find bounds on the average-case behavior of several on-line bin packing problems. For example, the number of unmatched points in an up-right bin packing problem is equal to the amount of empty space produced the Best Fit algorithm. The rightward matching problem gives lower bounds for any on-line bin packing algorithm. Bounds on the performance of the First Fit bin packing algorithm can be obtained by looking at a planar matching problem with an extra condition added on pairs of edges.

The problem of bin packing is: given n items of sizes between 0 and 1, fit them into the least number of bins such that the sum of the sizes of the items in any bin does not exceed 1. This is an NP-complete problem that has received much study.

One topic of study has been the behavior of simple algorithms for bin packing. Some of the first results, proved by Johnson et al [Jo,JDUGG] were that in the worst case, the First Fit algorithm could not use more than $(17/10)\mathrm{OPT}+1$ bins and First Fit Decreasing could not use more than $(11/9)\mathrm{OPT}+4$ bins, where OPT is the number of bins used by the optimal packing. Several algorithms have been found since that improve on these results. Recently, asymptotically optimal algorithms have been found [FL,KK] where the ratio of bins used to the optimal number approaches 1 as the number of items goes to infinity. The best asymptotic result so far is that of Karmarkar and Karp [KK], who give an algorithm that never uses more than OPT + $\log^2$ OPT bins.

Work has also been done on average-case analysis. For the average case, one must assume some distribution on the item sizes. Much of the work done on this problem has assumed that the item sizes are uniformly distributed on $[0,1]$. For this distribution, Coffman, Hofri, So and Yao [CHSY] showed that the expected ratio between the algorithm's performance and the optimal packing was 4/3 for the algorithm Next Fit. More recently, Frederickson [Fr] showed that for First Fit Decreasing, this ratio approaches 1 as the number of items goes to infinity. Lueker [Lu] then showed that the expected wasted space for this algorithm is $\Theta(n^{1/2})$, where wasted space is the total amount of empty space in partially filled bins, and n is the number of items. The latest results along these lines are in Bentley et al [BJLMM]. They show that the expected wasted space for the algorithm First Fit is $O(n^{4/5})$.

There are also results for other distributions. Karmarkar [Ka] analyzed Next Fit for a uniform distribution on $[0,\alpha]$. Karmarkar, Karp, Lueker and Murgolo [KKLM] have generalized Lueker's result of $\Theta(n^{1/2})$ wasted space for First Fit Decreasing to any symmetric or decreasing distribution. Recently, Bentley et al [BJLMM] have shown the surprising result that First Fit Decreasing packing items uniformly distributed on $[0,\alpha]$, $\alpha \leq \frac{1}{2}$ produces with high probability *constant* wasted space.

We will concentrate on on-line algorithms packing items uniformly distributed on $[0,1]$. On-line algorithms are algorithms, for example Next Fit or First Fit, that assign items to bins as the items are input. For some applications, on-line algorithms are necessary. Unfortunately, unlike off-line algorithms, on-line algorithms can never achieve asymptotically optimal worst-case performance. It has been shown [Br,Li] that in the worst case, any on-line algorithm must use at least 1.536 times as many bins as the optimal packing.

We investigate the average-case behavior of on-line algorithms. We first show, in Chapter 5, that the algorithm Best Fit, given items from a uniform distribution on $[0,1]$, is equivalent to the planar up-right matching problem discussed in Part I of this thesis. The bound of $\Theta(n^{1/2}\log^{3/4}n)$ thus applies to the wasted space produced by Best Fit. In Chapter 6, we show that First Fit is equivalent to a different planar matching problem, and analyze this problem to obtain bounds of $\Omega(n^{2/3})$ and $O(n^{2/3}\log^{1/2}n)$. In Chapter 7, we show that any on-line algorithm that does not know in advance the number of items must produce expected wasted space $\Omega(\sqrt{n\log n})$ when packing items uniformly distributed on $[0,1]$. This contrasts with off-line algorithms (or on-line algorithms that are given the number of items in advance) such as First Fit Decreasing, which can produce $\Theta(n^{1/2})$ wasted space.

The reason that on-line bin packing problems with items taken from a uniform distribution on $[0,1]$ correspond with planar matching problems is that in the optimal packing of items with this distribution, almost all bins are packed with two items. Thus, we need only find a matching between items larger than $\frac{1}{2}$ and items smaller than $\frac{1}{2}$ to find a near-optimal packing. Although the algorithms we analyze, First Fit and Best Fit, do not exclusively pack items smaller than $\frac{1}{2}$ with items larger than $\frac{1}{2}$, most of the bins they pack are packed in this manner. We will show that Best Fit finds a near-optimal matching and analyze first fit by looking at the matching that it finds.

Throughout this part we will mean by high probability, with probability at least $1-\frac{1}{n}$. In bin packing, anything with probability lower than $\frac{1}{n}$ can be ignored in analyzing expected case behavior. This is because n items can never take more than n bins to pack. Thus, if a bin packing algorithm packs items using $O(f(n))$ bins with probability $1 - \frac{1}{n}$, the expected number of bins used is

$$\left(1 - \frac{1}{n}\right) O(f(n)) + \frac{1}{n} O(n) = O(f(n)).$$

We will be measuring the performance of bin packing algorithms both by number of bins used and by wasted space. The wasted space is the total amount of empty space in bins, so the amount of wasted space is the number of bins minus the sum of the sizes of the items. The expected item size is $\frac{1}{2}$, so we have

$$\mathrm{E}(\text{number of bins}) = \frac{n}{2} + \mathrm{E}(\text{wasted space}).$$

For some proofs, the number of bins is a more convenient measure, while for other proofs, wasted space is more convenient.

Chapter 5. Best Fit

5.1. Introduction

We now discuss the Best Fit (BF) algorithm. In this algorithm, each item is placed into the fullest bin in which it fits at the time of arrival. We will use two variations of Best Fit, which we call 2-Best Fit (2BF) and Matching Best Fit (MBF). The algorithm 2BF uses the same rules as BF, except that it may never place more than two items in any bin. That is, it places each item in the fullest bin it fits in that contains only one item; failing this, it puts the item in an empty bin. For MBF, we impose this constraint (at most two items per bin) and also the constraint that an item of size less than $\frac{1}{2}$ may never have an item placed on top of it. That is, any bin containing an item of size less than one half is considered full. In this chapter, we will give bounds on the expected space wasted by Best Fit.

Theorem 5.1.1: The expected wasted space produced by the algorithm Best Fit when packing n items uniformly distributed on $[0,1]$ is $\Theta(n^{1/2}\log^{3/4} n)$.

We will first prove the upper bound and then the lower bound. These proofs use the theorems on matching in a plane that we introduced in Part I of this thesis. In Section 5.2 we show that the wasted space produced by Best Fit is bounded above by the up-right matching problem discussed in Part I of this thesis. The following section shows Best Fit is also bounded below by up-right matching, and is thus within a constant factor of it. Using the bounds for up-right matching, these results give that Best Fit wastes $\Theta(n^{1/2}\log^{3/4} n)$ space.

5.2. The Upper Bound

We first show the upper bound on wasted space. This proof involves two parts. First we show that MBF never uses fewer bins than BF. Second, we show that MBF is equivalent to the up-right matching problem. The $O(n^{1/2}\log^{3/4} n)$ upper bound for this matching problem thus also applies to the wasted space in the BF algorithm.

To prove that MBF always uses at least as many bins as BF, we need a lemma. Suppose that L is a list of items. We denote the packing of L using algorithm A by $A(L)$, and the number of bins used in this packing by $\#A(L)$. The lemma follows. We will be using several versions of it throughout this paper.

Lemma 5.2.1: If L' is a list obtained by removing one item from L, then

$$\#2BF(L) \geq \#2BF(L') \geq \#2BF(L) - 1$$

and

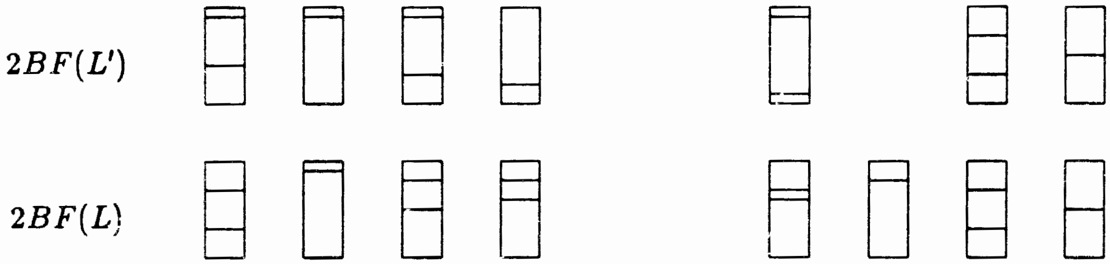
$$\#MBF(L) \geq \#MBF(L') \geq \#MBF(L) - 1.$$

Proof: We will prove the result for $2BF$. The exact same proof works for MBF. We show that, if we consider any bins with two items in them (and thus "full") to be identical, then L and L' differ in at most one bin. In fact, they are related in one of the following ways:

A) $2BF(L)$ can be obtained from $2BF(L')$ by replacing a 1-item bin by a full (2-item) bin.

B) $2BF(L)$ can be obtained from $2BF(L')$ by adding a 1-item bin.

We prove Lemma 5.2.1 by induction. We must show that if we add an item p to two packings related by A or B, they will still be related by A or B.



Packings related by A. Packings related by B

Figure 5.2.1:

Suppose that $2BF(L)$ and $2BF(L')$ satisfy A. Then there is a 1-item bin in $2BF(L')$ that is not in $2BF(L)$. Call this bin b. If the next item to be added, say p, does not go into the bin b, then the two packings will still be related by A since the item p goes into the same bin in both of them. If the added item goes into bin b in $2BF(L')$, then in $2BF(L)$ it will either go into an empty bin or it will go into a different 1-item bin. In the first case, the two packings will be related by B. In the second case they will be related by A (See Figure 5.2.1.)

A similar analysis holds if $2BF(L)$ and $2BF(L')$ are related by B. Let b be the 1-item bin in $2BF(L)$ not in $2BF(L')$. If the next item, say p, does not go into b, the packings will still be related by B. If p does go into b, then in the $2BF(L')$ packing it wither goes into an empty bin or into a 1-item bin. In the first case, the packings are related by A, and in the second, by B.

We must still do the base case of the induction. For this, we must show that when the item p in L' not in L arrives, the two packings will be related by either A or B. Before this item arrives, $2BF(L) = 2BF(L')$. The item p must either go into a bin by itself, or into a 1-item bin. In the first case, the packings are related by B; and in the second case, they are related by A.

The proof for MBF is exactly the same; simply replace 2BF in the above paragraphs with MBF. With MBF, if the 1-item bin mentioned in relation A or B contains an item with size less than $\frac{1}{2}$, then the bin is considered full and will never receive another item. This fact does not affect the proof. ∎

We can now prove the following lemma:

Lemma 5.2.2: If L is a list of items, then

$$\#MBF(L) \geq \#BF(L).$$

Proof: We first show that $\#MBF(L) \geq \#2BF(L)$, and then show that $\#2BF(L) \geq \#BF(L)$. Obtain a list L' by removing from L all items of size less than $\frac{1}{2}$ which were packed in a bin by themselves in $MBF(L)$. Because these items do not affect the way the other items are packed, the algorithm MBF will not put any item of L' smaller than $\frac{1}{2}$ in an empty bin. The items of L' will thus be packed in the same way by $2BF$ and MBF. This gives

$$MBF(L') = 2BF(L').$$

Also, since the items we removed were packed one per bin by MBF,

$$\#MBF(L') + |L - L'| = \#MBF(L).$$

Since L' was formed by removing items from L, Lemma 5.2.1 implies that

$$\#2BF(L') + |L - L'| \geq \#2BF(L).$$

Together, these equations show that

$$\#MBF(L) \geq \#2BF(L).$$

The proof that $\#2BF(L) \geq \#BF(L)$ is similar. The same proof for First Fit is contained in [BJLMM]. As before, we define a sublist L' by removing items from L. For every bin of $BF(L)$ containing more than two items, remove from L all items packed after the second item. This gives L'. We removed all items packed more than two per bin in $BF(L)$, so $2BF(L')$ packs the items of L' in exactly the same way that $BF(L)$ packs the items of L', and in $BF(L)$ the items in $L - L'$ are packed into bins containing at least two items from L'. Thus, we get

$$\#2BF(L) \geq \#2BF(L') = \#BF(L).$$

The inequality here follows from Lemma 5.2.1, since L' was derived from L by removing items. Combined with the previous inequality, this proves the lemma. ∎

Now we are ready to convert the problem to up-right planar matching. The up-right planar matching problem is: given +'s and −'s uniformly distributed in a unit square, match as many −'s to +'s as possible, with the constraint that a − can only be matched with a + above and to the right of it. Karp et al used this problem to obtain bounds on 2-dimensional bin packing [KLM]. Here, by considering time as a second dimension, we use it to obtain bounds on 1-dimensional bin packing for the Best Fit algorithm.

To convert a list of items to a planar matching problem, we represent the items received by the algorithm by points in a unit square. The x-coordinate will be the size of the item. The y-coordinate will depend on the time the item was received. To fit the n items into the square, we put the jth item at distance j/n from the top. Next, we label the items larger than $\frac{1}{2}$ with '+' and those smaller than $\frac{1}{2}$ with '−'. We then fold the plane about the line $x = \frac{1}{2}$ (See Figure 5.2.2), so a + point with x-coordinate s will be moved so it has x-coordinate $1 - s$. For every bin in MBF containing two items, we will put an edge between these items. This gives a bipartite matching M between + and − points.

We claim every edge in a MBF matching M matches a − to a + above and