Analysis of the Unfair First Fit Algorithm

John Leung

CS530 – Advanced Algorithm Design and Analysis

Cal Poly Pomona

Fall, 2002

Abstract. The authors of the paper, Azar, Boyar, Epstein, Favrholdt, Larsen, and Nielsen proposed an on-line *Unfair First Fit* Bin Packing algorithm[1] that can maximize the number of items packed in bins. More specifically, the authors claimed that the *competitive ratio*, that is the ratio of number of items packed by an on-line algorithm to the number items packed by an off-line algorithm, is greater than any other on-line packing algorithms. This analysis will discuss and comment on the authors' approach, method, and usefulness of their algorithm.

1. Introduction

According to [2], imports and exports accounts for 30 percent of the US economy. Of course, goods have to be packed into cargos before they can be imported or exported. Thus, packing is a very important aspect of this import/export economy. If there is a better way to pack these shipments, billions of dollars can be saved. And of course, the algorithm that enables this will be so valuable that it may be considered priceless. Is this on-line *Unfair First Fit* Bin Packing algorithm the one? Well, personally, I think not. It is too general and requires too much extra work, both of which I will address in this analysis. But it does provide some basis where further developments based on it can that Holy Grail!

2. Background Information

Before researching this paper, I was not exactly sure what the word "on-line" means. The best definition I got from searching over the Internet is "Decisions must be made with out any knowledge of future requests." Applying this to the packing algorithm would probably mean that items to be packed are coming in one at a time, and decision to what to do with it must be made without knowing the size of the next one. Where as an off-line packing algorithm would be the items to be packed can all be seen and so decisions on how to pack them can be better made.

In the Dual Bin Packing problem, which authors' paper is based on, only a limited number of (n) bins are given to pack a set of items. In the on-line **Fair** First Fit algorithm, the items come in one at a time and if it fits in the current bin, it will be packed, if it doesn't fit, then pack it in the next bin. After all n bins are packed, the items that are left unpacked are considered to be Rejected (R). The measure of the quality of these on-line algorithms is the *competitive ratio*, as stated in abstract, is the ratio of number of items packed by an on-line algorithm to the number items packed by an off-line algorithm. To make the computations easier, an optimal off-line algorithm that can always pack all items in n bins is considered. The competitive ratio of Fair First Fit algorithms is shown in other papers to be 5/8, or more precisely $5/8 + O(1/n^{1/2})$. (Note that this is not the complexity of the algorithm, but the bound of the competitive ratio, which is considered to be more important than the complexity in these problems.) The competitive ratio of the proposed *Unfair First Fit* algorithm is $2/3 + O(1/n)$. For large n's, this is approximately 4.2% difference.

3. The algorithm

The *Unfair First Fit* algorithm is rather simple. (In fact, it is too simple to put to any real use, more on this later in the discussions section.) The authors only considered the un-weighted version of the Bin Packing problem, that is all the items have the same value, no matter what the sizes were. Therefore, in order to fit the most items, it is better to reject larger items rather than trying to fit them in the first place (this is where the word *Unfair* comes from). The cutoff line between a ‘large’ and a ‘small’ item is 1/2. If the item is larger than 1/2, and the performance ratio would still be at least 2/3 even if the item were rejected, then it will be rejected. Below is the pseudocode(comments are added for easier understanding):

Input :

  $S = <o_1, o_2, \ldots, o_n>$

Output :

  A // set of accepted items; R // set of rejected items; and a packing of those items in A

```
A := { };  R := { }
While S != <>
        O := head(S)
        If  (size(o) > 1/2) AND (|A| / (|A| + |R| + 1) >= 2/3 )
                R := R Union {o}
        Else if there is space for o in some bin
                Pack o according to the First Fit rule
                A := A Union {o}
        Else
                R := R Union {o}
End while
```

After all the items are accepted or rejected, it will rearrange the packed items (transferring them from one bin to another) to create room for rejected items. The rejected items are then packed as much as possible.

4. Analysis/Discussions

The complexity of the algorithm is just O(n).

The running time function is:

$$T(n) = \{ \quad 1 \qquad n = 1$$
$$\qquad T(n) + 1 \qquad n > 1 \}$$

However, as mentioned before, the main bounds of this type of problem is not the complexity, but the competitive ratio. The competitive ratio of the *Unfair First Fit* algorithm is 2/3 + *O*(1/n), which is just 2/3 when n is big. To get this magic number 2/3, the authors used other magic numbers 1/3 and 1/2 (OK, I'll let the 1/2 go since the only reasonable cutoff between "small" and "big" is 1/2) in their algorithm and proofs. The proofs are sound, but it's like the chicken and egg problem, they used 1/2 and 1/3 (e.g. in Theorem 4.1, case 1: size > 1/2 , … case 2a: size <= 1/3, …) to obtain the competitive ratio 2/3. If these numbers were came up by trial and error, at least state so.

Well, this is actually a minor flaw, I believe, when comparing to the other problems. First, as I mentioned before, the algorithm is too general to be put to real use. Items do not come in one-dimensional sizes. An item can be a cube and have a volume of less than 1/2, yet take up more than "half" of the space. Yes, it is not too hard to just add three-dimensional calculations to the algorithm. However, I doubt doing so will yield the same competitive ratio, which will make the whole premise of the algorithm false! To have the algorithm stand correct, it should be titled "linear packing", not "bin packing".

Secondly, if the items are really non-weighted in the real world, I am sure people would be rejecting bigger items too. This problem is not too bad; a simple change in the condition statement should correct this:

Change:

If (size(o) > 1/2) AND (|A| / (|A| + |R| + 1) >= 2/3)

To:

If (size(o) > 1/2) AND (|A| / (|A| + |R| + 1) >= 2/3) AND (o's weight < pre-defined threshold)

Third, to make the algorithm more ideal, the authors should base the algorithm on the Classical Bin Packing model where the number of bins is not limited. True, the number of bins cannot really be "unlimited", but which shipper will dare to run out of bins before the items run out? Off-line is the only situation that it is more useful to set a limit on the number of bins to optimize the packing.

Finally, when I find out that the author used the idea of re-packing the items to make room for the rejected items, I was very disappointed. The time it takes to rearrange the items will take anywhere from zero time(if no items have to be rearranged) to twice as long(if all items have to be rearranged) as the packing itself! Think about the time it will take (even machines) to rearrange items that are few thousand pounds. The whole idea of the on-line algorithm is that decisions need to be made right there, rearrangements should only be done off-line.

5. Conclusion

The title and abstract of the paper got me very excited in learning what new approach they came up with to pack 4.2% more items in the on-line bin packing problem set. After really analyzing it, I was deeply disappointed with the solution they came up with. I can live with most of the problems stated above, but the idea of rearranging the items to get that extra 4.2% packed I don't. And I do not see many situations where this can be more beneficial than the Fair First Fit algorithm, but it does provide some idea for further development.

References

[1] Y. Azar, J. Boyar, L. Epstein, L. Favrholdt, K. Larsen, and M. Nielsen. *Fair versus Unrestricted Bin Packing* pages 181-196, Algorithmica, Oct. 2002.

[2] Department of State, International Information Program, http://usinfo.state.gov/journals/ites/1000/ijee/trans-slater.htm