Analyzing Bin Packing Algorithms

Byron Fong

Introduction

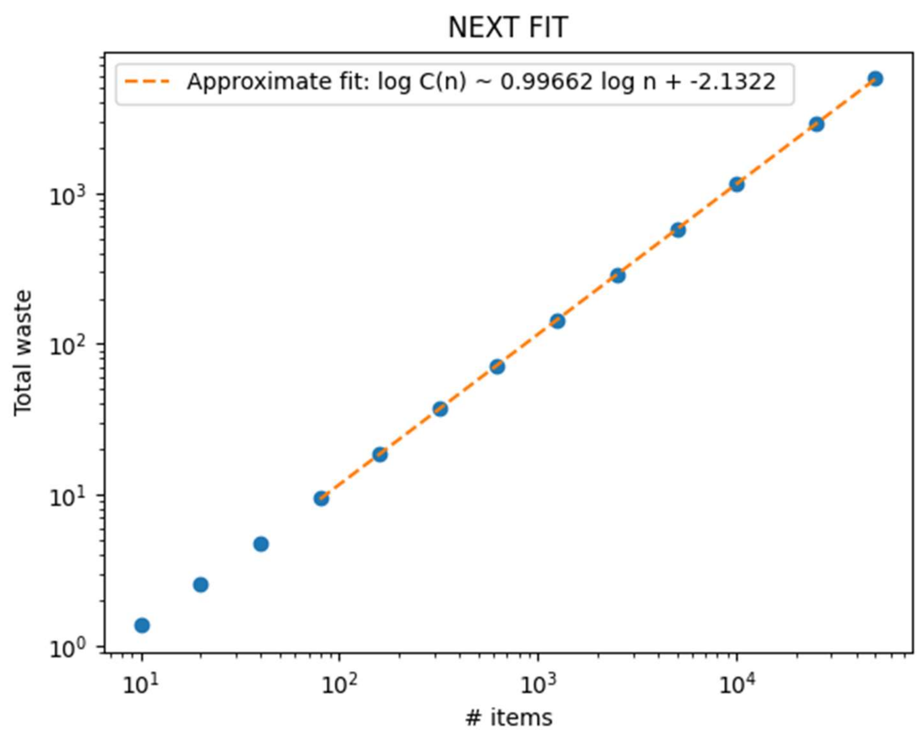
This investigation aims to test several bin packing algorithms to determine which one produces the highest quality solution to the bin packing problem – that is, the solution that minimizes waste for any combination of input. Five different bin packing algorithms were tested: Next Fit (NF), First Fit (FF), Best Fit (BF), First Fit Decreasing (FFD), and Best Fit Decreasing (BFD).

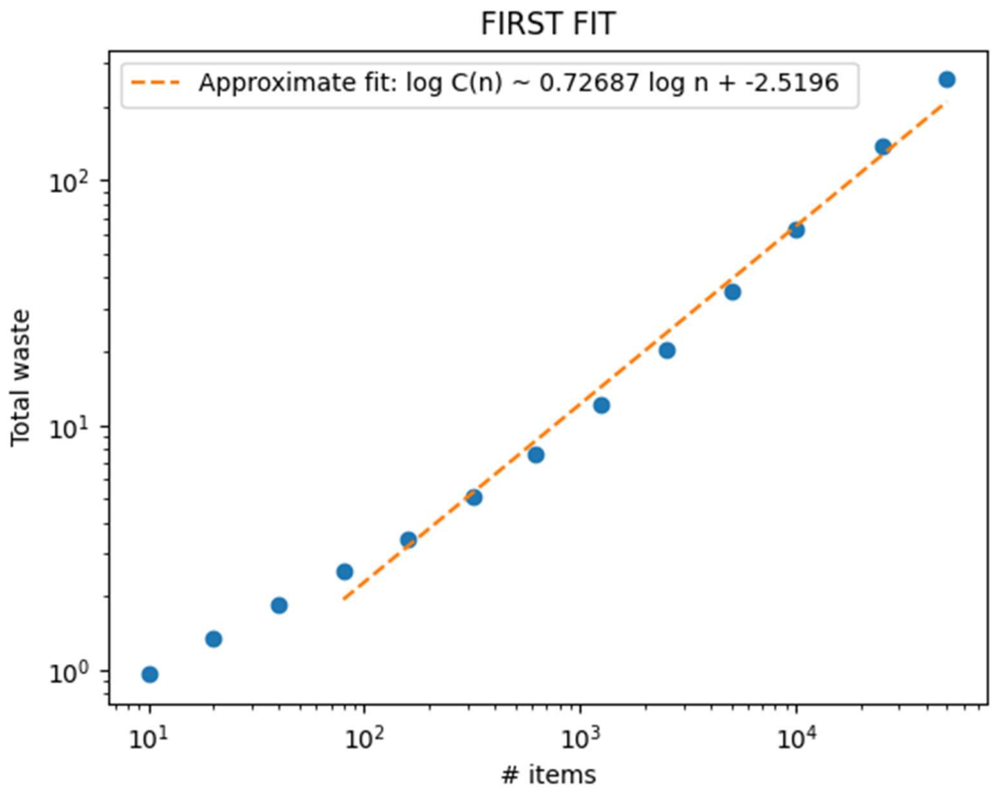
Procedure

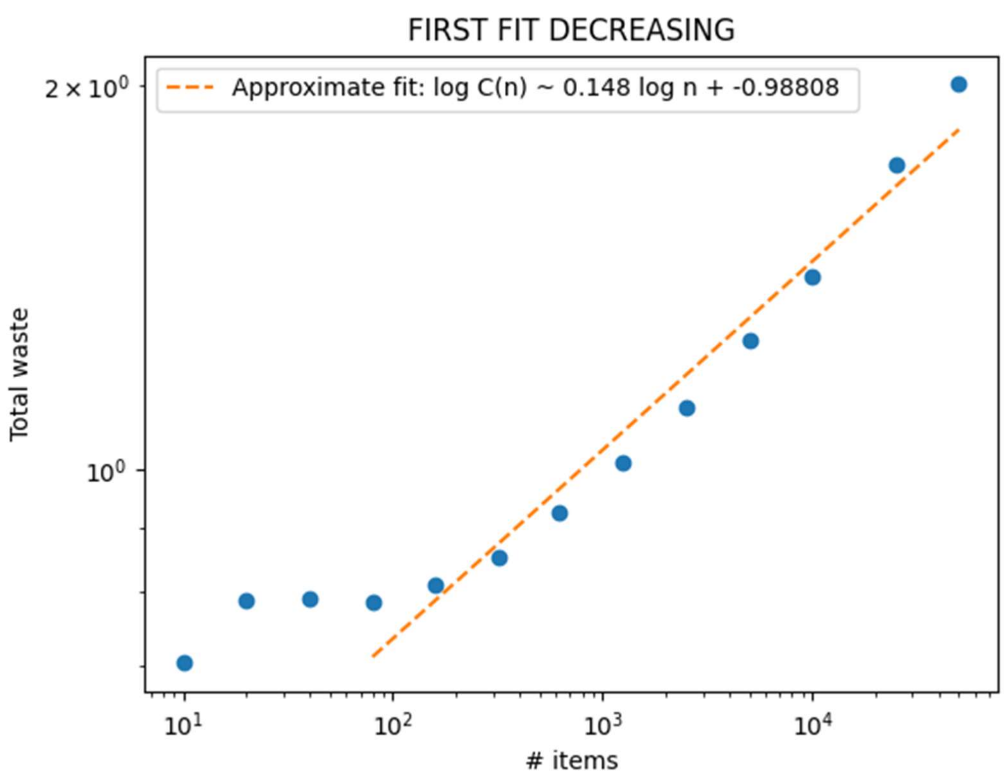
This instance of the bin packing problem was tested with bins of container size 1, and various items to be placed in the bin with values between the sizes 0.0 and 0.7. These values were chosen over a uniform distribution. The number of items tested ranged from 10 to 50000. For each input, one of the bin packing algorithms would be executed with it, and the waste measured after its execution. Waste was calculated from taking the total amount of space (used or unused) in the number of bins used, minus the sum of all the items in the original input. The resulting data, number of items vs. waste remaining, was then plotted on a log-log graph.

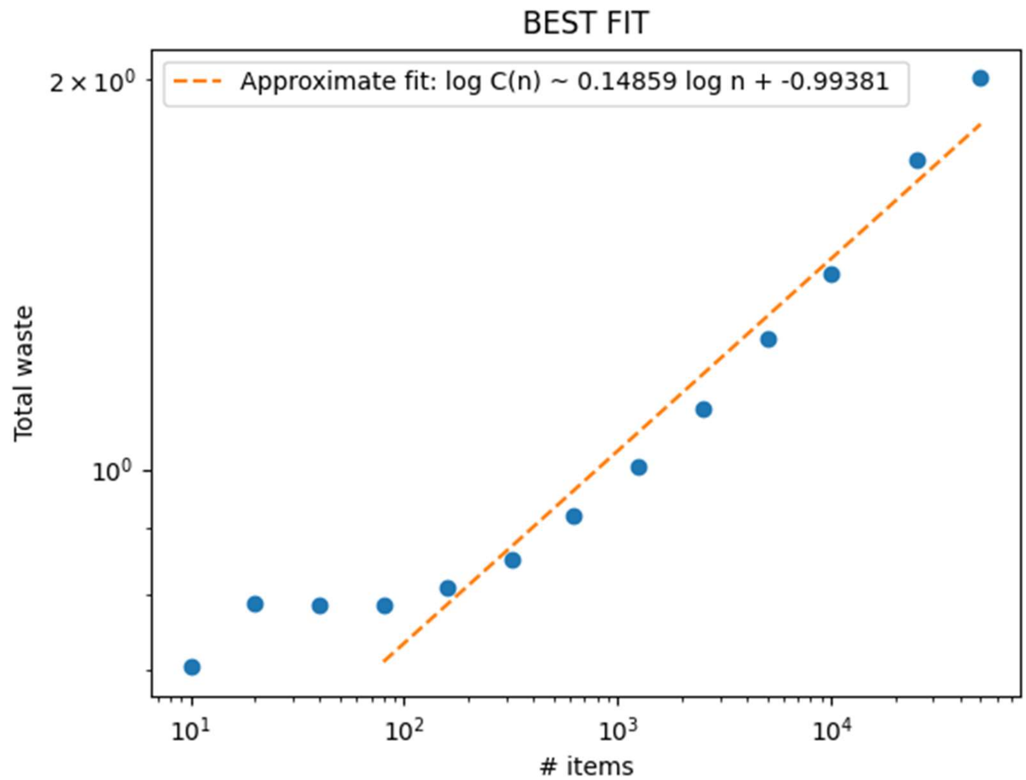
Findings

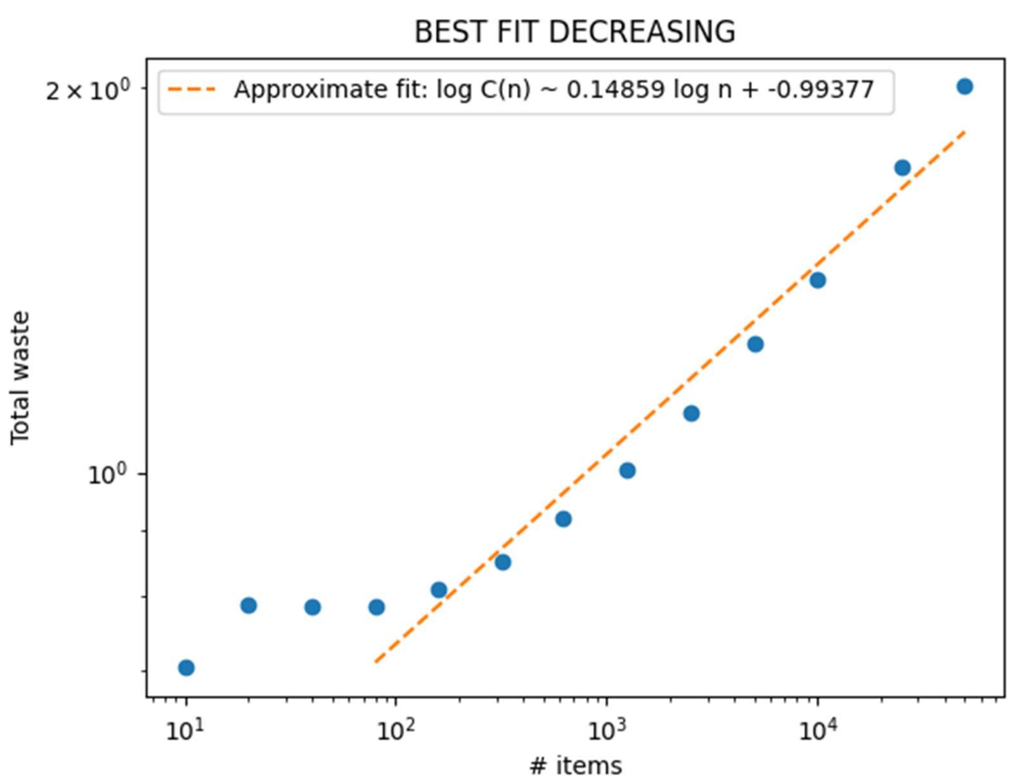
Following are the graphs of the five different bin packing algorithms, with best fit lines plotted to them.




FIRST FIT
Approximate fit: log C(n) ~ 0.72687 log n + -2.5196
Total waste
items


FIRST FIT DECREASING
Approximate fit: log C(n) ~ 0.148 log n + -0.98808
Total waste
items


BEST FIT
Approximate fit: log C(n) ~ 0.14859 log n + -0.99381
Total waste
items


BEST FIT DECREASING
Approximate fit: log C(n) ~ 0.14859 log n + -0.99377
Total waste
items

Observations

The approximate function W(n) as a function of n for each of the algorithms are approximately:

-Next Fit: $\log W(n) = 0.996 \log n – 2.132$

-First Fit: $\log W(n) = 0.727 \log n – 2.520$

-First Fit Decreasing: $\log W(n) = 0.148 \log n – 1$

-Best Fit: $\log W(n) = 0.149 \log n - 1$

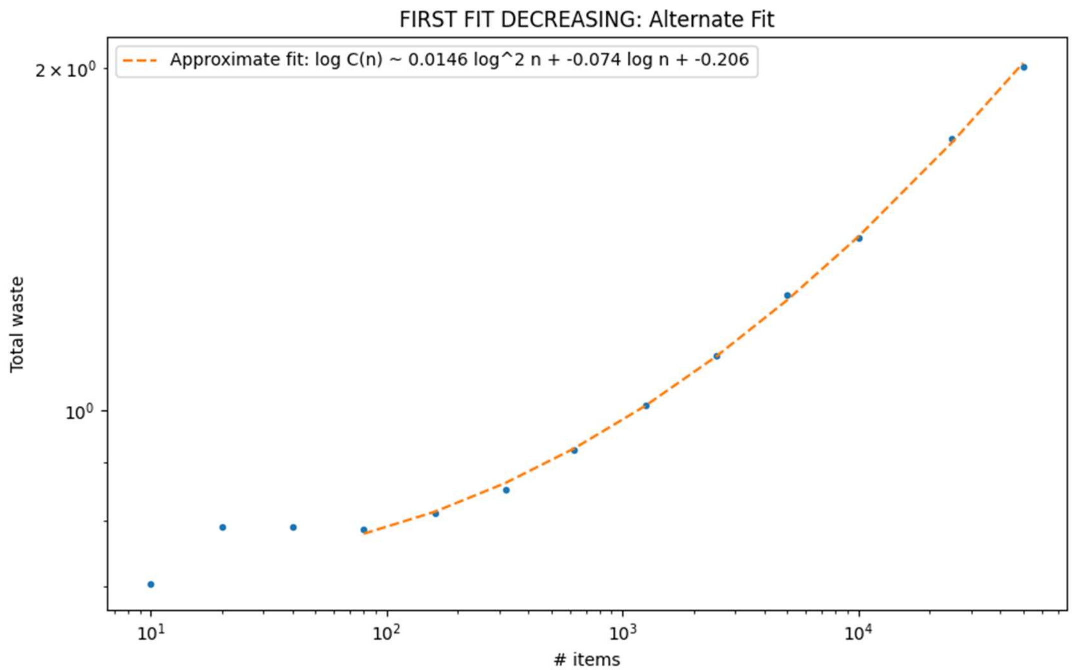
-Best Fit Decreasing: $\log W(n) = 0.149 \log n – 1$

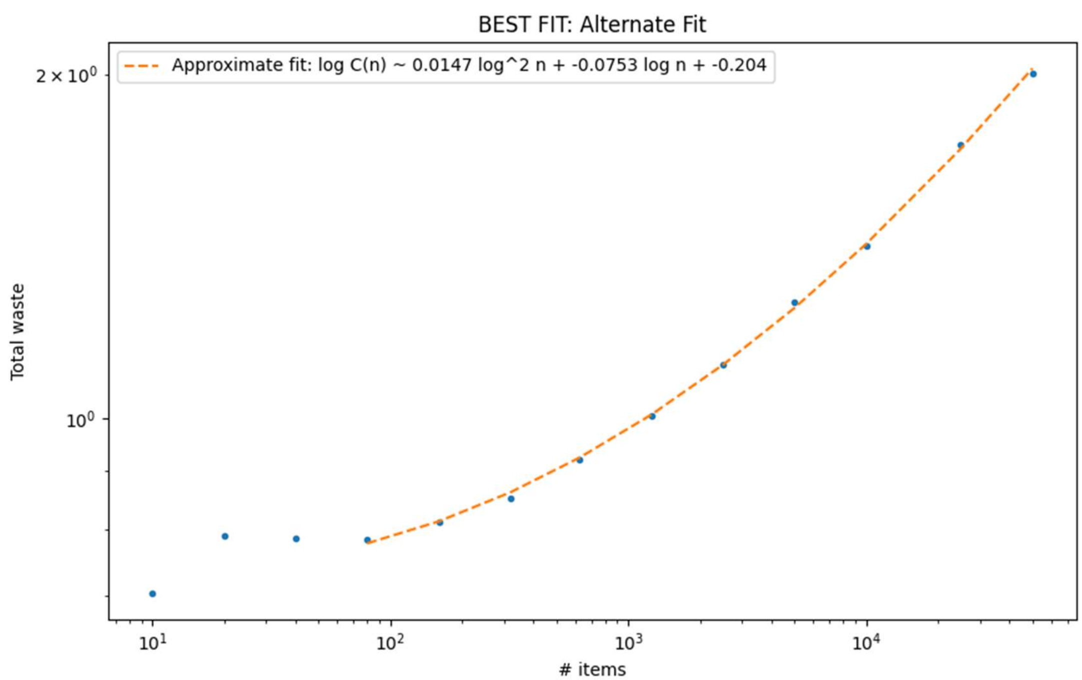
The Next Fit packing algorithm has the fastest growing waste as a function of items, with its waste function growing linearly with a slope of ~1. First Fit which has the second worst, with a slope of ~0.727. Note that there is a very slight curve in the data for First Fit, but it is negligible enough to have a reasonable linear fit.

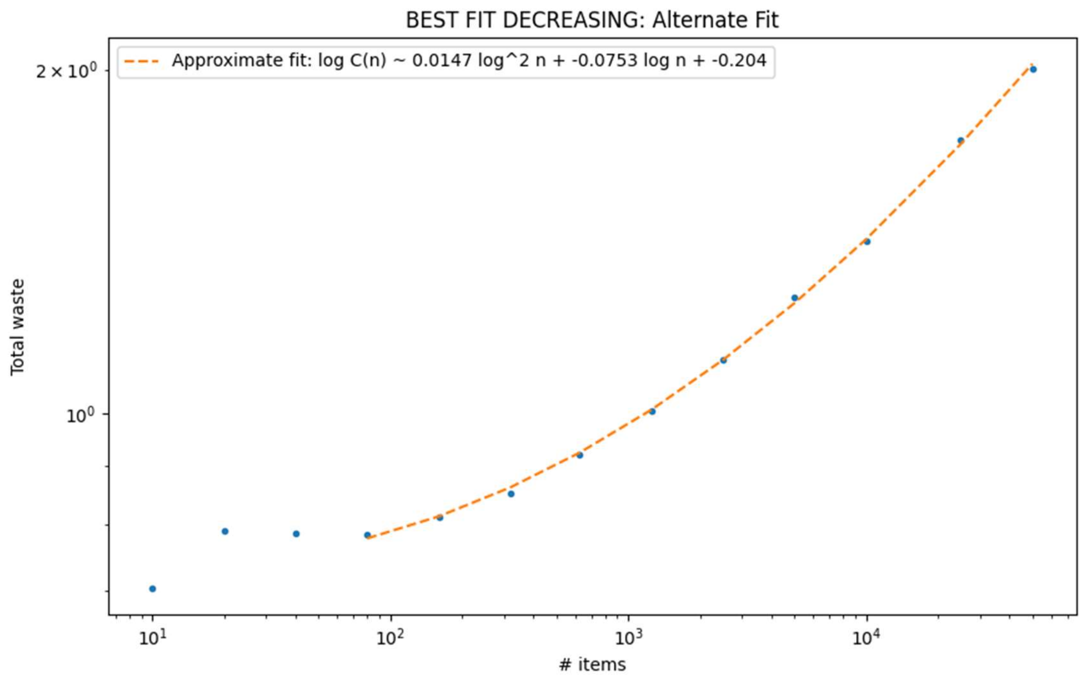
The First Fit Decreasing data has a slope of ~0.148, and there are a few peculiarities in the graph. Firstly, the data is noisy with small n: at some instances of n when n is small, the value of waste W(n) decreases as n grows. This is counter-intuitive, so instances of small n were not included when calculating the best-fit line. Secondly, the best fit line seems to extrapolate negative values of waste for smaller n. This is due to the best fit line being asymptotic, as it only extrapolates accurate waste values for bigger n. Thus, the best fit line for First Fit Decreasing is inaccurate for small numbers of n and should not be used as such.

The Best Fit Decreasing data and Best Fit Decreasing data is extremely similar, both having slopes of ~0.149. The data comes with the same problems as first fit decreasing, being both noisy and having a best-fit line that extrapolates negative values of waste for small n.

When reviewing the graphs for some of the algorithms, namely FFD, BF and BFD, I noticed that a polynomial fit would fit the data more strongly. Below are the graphs with polynomial functions to a degree of 2 fit to them:


FIRST FIT DECREASING: Alternate Fit
Approximate fit: log C(n) ~ 0.0146 log^2 n + -0.074 log n + -0.206
Total waste
items


BEST FIT: Alternate Fit
Approximate fit: log C(n) ~ 0.0147 log^2 n + -0.0753 log n + -0.204
Total waste
items

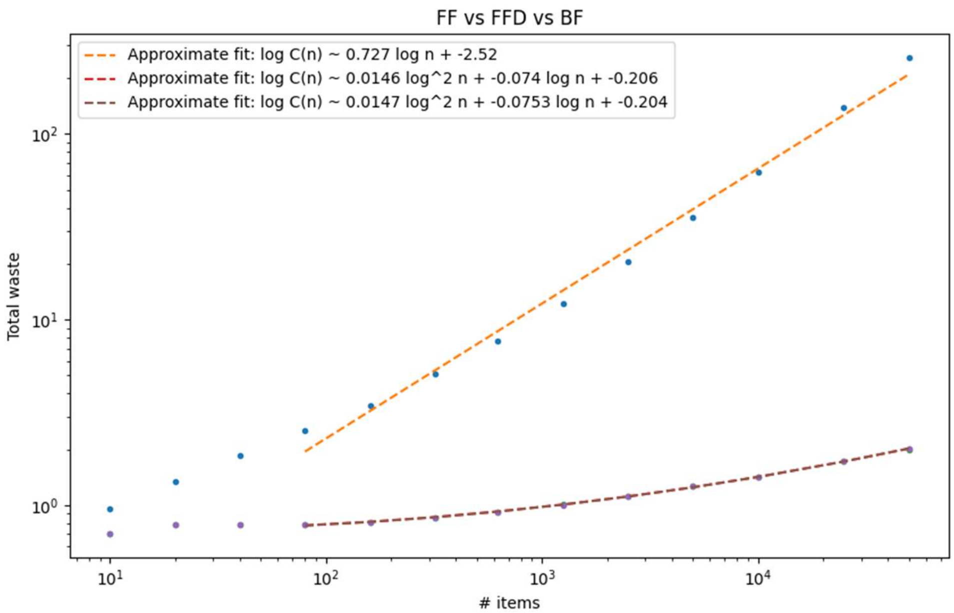


The new approximate waste functions for these algorithms are therefore:

-First Fit Decreasing: $\log W(n) = 0.0146 \log^2 n - 0.074 \log n - 0.206$
-Best Fit: $\log W(n) = 0.0147 \log^2 n - 0.0753 \log n - 0.204$
-Best Fit Decreasing: $\log W(n) = 0.0147 \log^2 n - 0.0753 \log n - 0.204$

Analysis

The Next Fit and First Fit algorithms are clearly much worse than the others, with their growth rate of waste being more than five times the amount of the other algorithms. The remaining three algorithms, First Fit Decreasing, Best Fit, and Best Fit Decreasing, produce much less waste, and have an approximate waste function that is extremely similar to the other two in that group. A very peculiar observation I made is that when examining specific inputs for Best Fit and Best Fit Decreasing, I found that most, if not all, of the waste generated by the algorithms for those sets of inputs were the exact same.



Graph of First Fit, First Fit Decreasing, Best Fit. First Fit Decreasing and Best Fit generate much less weight than First Fit and the difference between FFD/BF is close to negligible.

Conclusion

The "best" bin packing algorithm is very close between First Fit Decreasing, Best Fit, and Best Fit Decreasing, but it would be First Fit Decreasing by an extremely small margin. This algorithm has the lowest growth rate in terms of waste out of all other algorithms, with a leading coefficient of 0.148 with a linear fit. The intuition for First Fit Decreasing being the most efficient algorithm is there as well – the algorithm will keep adding new bins for the first few items until the value of the items reach around ~0.5, at which point the algorithm will find the newest bins at the end and add items to them accordingly. With the remaining items, the algorithm would keep inserting to already existing bins, from newer to older bins as the newer ones reach around a total of space of 1 and become full.