Adarsh B Shankar
ID: 11972805
CS 165 – Spring 2021

Project 2: Bin Packing Algorithms

Overview:

Project 2 involves the testing quality of the various bin packing algorithms, the algorithms tested in this project are as follows:

1. Next Fit (NF)
2. First Fit (FF)
3. First Fit Decreasing (FFD)
4. Best Fit (BF)
5. Best Fit Decreasing (BFD)

Bin Packing Problem:

Given n items with sizes $s_1,s_2,\ldots,s_n$ such that $0<s_i<1$ for all $1\leq i\leq n$, we want to pack them into the fewest number of bins with all capacity of 1. This problem is NP-hard, no known polynomial time algorithm is known and conjectured that none exists.

Bin Packing Algorithms:

Next Fit: - Loop through each item. If the current item can fit in the current bin we place it there else we start a new bin.

First Fit: - Loop through each item. For each item we go through all the current bins and find the first bin that can occupy the current item. If none exist, we start a new bin.

First Fit Decreasing: - First sort items by size from largest to smallest. Then run the first fit algorithm.

Best Fit: - Loop through each item. For each item we go through all the current bins and find the bin that can occupy the current item the tightest (i.e. the smallest available space to fit in the current item). If none exist, we start a new bin.

Best Fit Decreasing: - First sort items by size from largest to smallest. Then run the best fit algorithm.

Adarsh B Shankar
ID: 11972805
CS 165 – Spring 2021

Experimental Testing:

Test Goal: The aim of this project is to test these different algorithms to determine an estimate for the waste W(N) of each algorithm as the number items N grows.

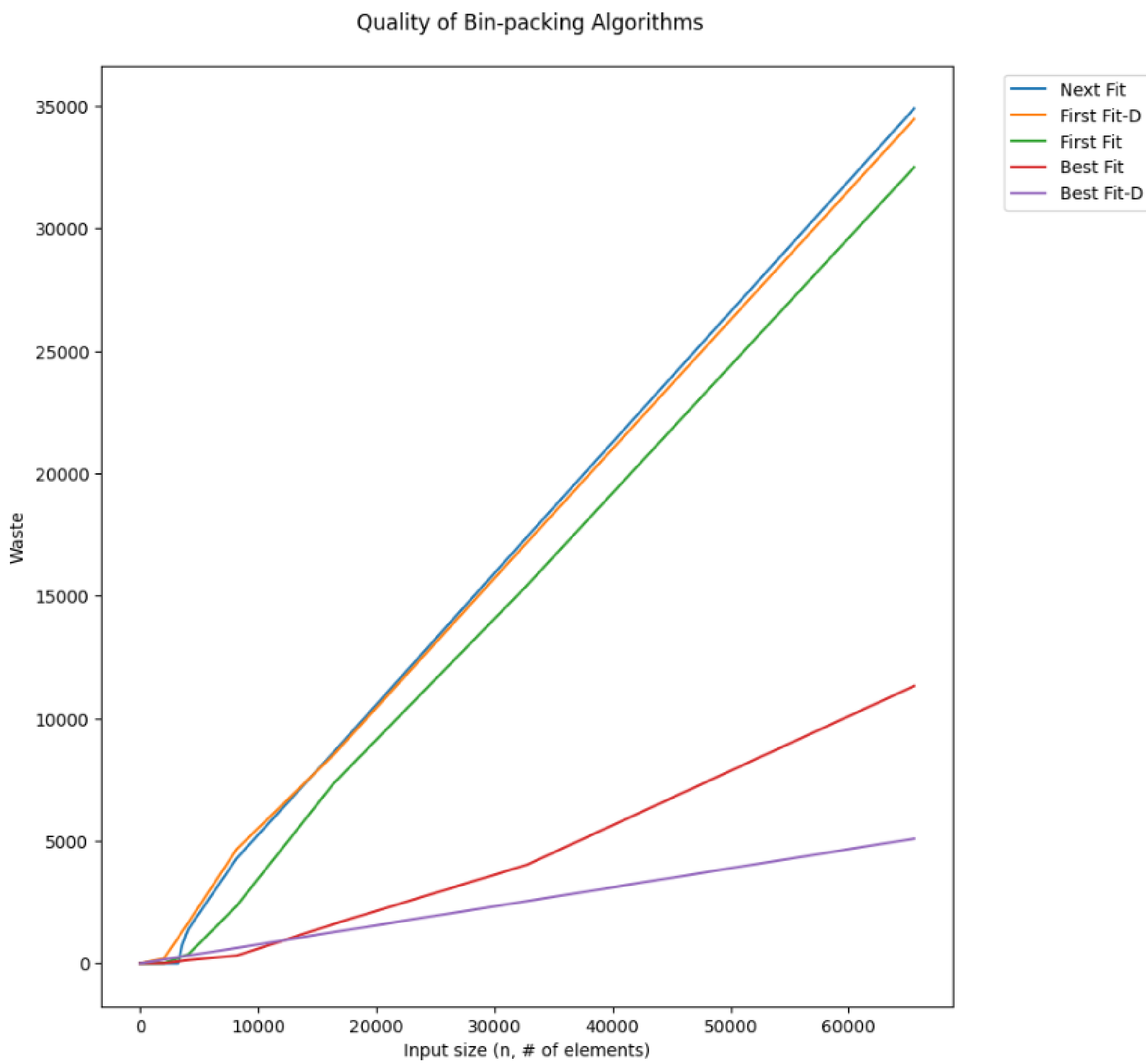
Testing Methods:

For each bin packing algorithm:

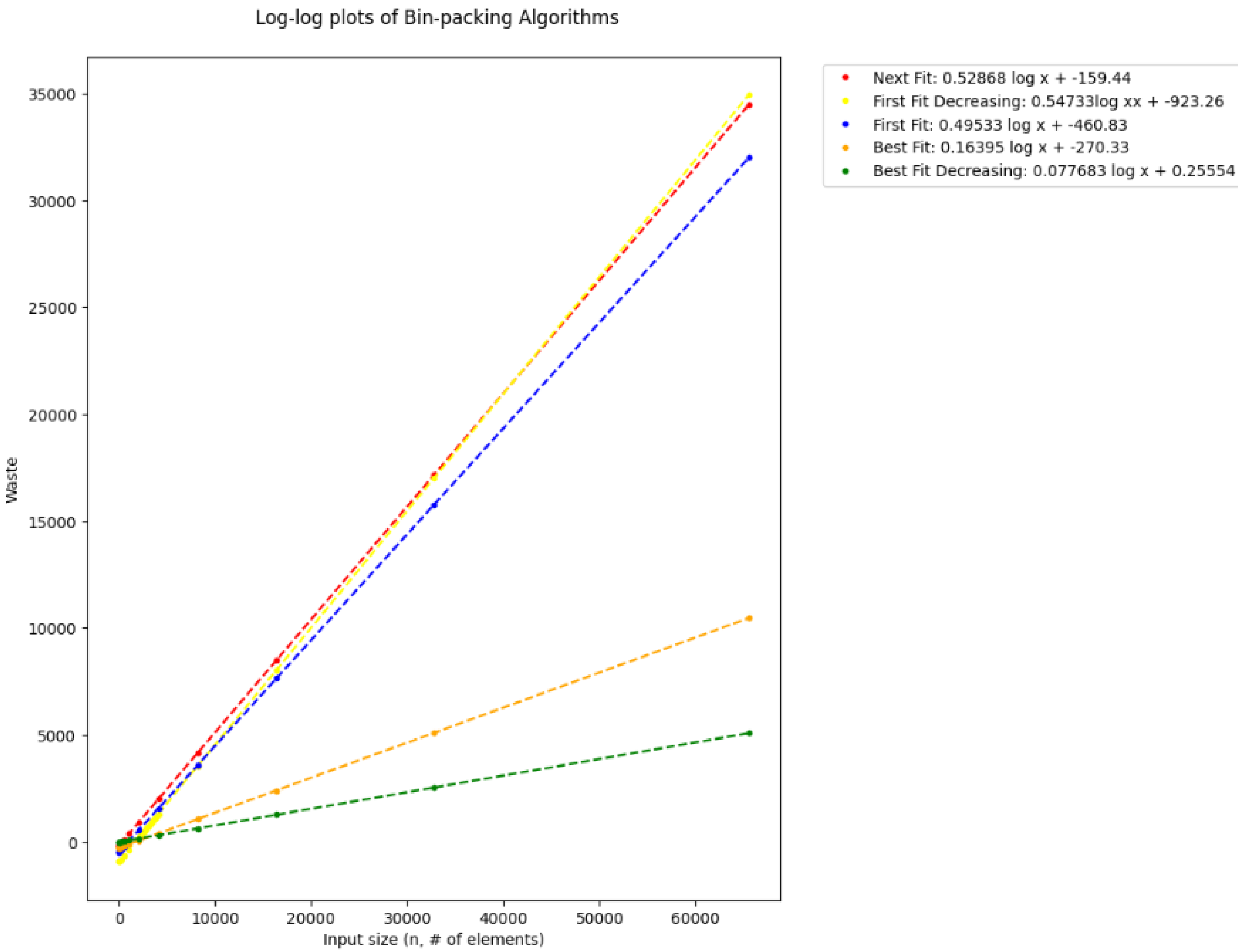
- Generate a uniformly distributed array/vector of items ranging from 0.0 – 0.6 for a size N.
- For this size we run the corresponding algorithm 10 times and average the waste generated by each run.
- We repeat this process for the algorithm for sizes ranging from $2^2 \leq N \leq 2^{16}$

Testing Results:

The graph below shows the various algorithms and see the waste generated as the size of items increase:



The graph below shows the log-log various algorithms and see the waste generated as the size of items increase:



<u>Testing Analysis:</u>

The above graphs also give the line of best fit for each log-log plot, so we can derive the log of the waste W(N) of each algorithm for each size N:

1. Next Fit (NF): W(N) = 0.528log(N) – 159.44 => Slope = 0.528
2. First Fit (FF): W(N) = 0.495log(N) – 460.83 => Slope = 0.495
3. First Fit Decreasing (FFD): W(N) = 0.547log(N) – 923.26 => Slope = 0.547
4. Best Fit (BF): W(N) = 0.164(N) – 270.33 => Slope = 0.164
5. Best Fit Decreasing (BFD): W(N) = 0.077log(N) – 0.255 => Slope = 0.077

Conclusion:

From the above graph, the best fit decreasing algorithm has the lowest slope, so this algorithm minimizes the waste as the input sizes increase the best. The best fit decreasing algorithm minimizes the waste best because the items are sorted in descending order and because the best fit algorithm alone is a greedy algorithm which optimizes the waste generated. However, the best fit algorithm alone produces a worse solution than the best fit decreasing algorithm because the decreasing algorithm will store the largest items in the respective bins first. Once the largest items have been stored, it is easier to pack smaller items tightly in the bins because the likelihood of the free space left in each bin being greater than or equal to the smaller items is much greater.