For lower bound, I used CSF + two different GFC. The first GFC is the costs of things that individually will not fit in the knapsack. The second GFC is derived from the sum of the cost of things that individually will not fit in the knapsack and the value of the objects with the smallest mass values selected from each group of objects that in total will not fit into the knapsack.

For upper bound, I used CSF + two different FFC. The first FFC is derived from the costs of things left on the table after applying a greedy heuristic on the remaining objects. The second FFC is from appending the values of objects remaining on the table from the end of a sorted array containing all the objects. In other words, the second way has the reversed order from the first way.

The amount partial solutions generated by the second lower bound method is 4 times less than the number of partial solutions generated by the first lower bound method in the file with size 20. In general, It greatly reduces the partial solution amounts and complexity.

However, there is not much of a big difference between using the first upper bound method and the second upper bound method. For size below 25, these two methods generate the same amount of partial solutions. After size 25, the second upper bound method seems to be slightly inefficient compared to the first one.