ANALYSIS OF VARIANTS OF THE SUBSET SUM ALGORITHM FOR BIN PACKING

Felicity Cunningham

Julian Mestre
School of Information Technologies
FACULTY OF ENGINEERING & INFORMATION TECHNOLOGIES

INTRODUCTION AND MOTIVATION

Bin Packing

Space is a valuable commodity in many situations, from storing belongings in boxes to allocating jobs to processors, or advertising during television breaks.

In all of these situations it is critical to optimize the use of space. The basic one-dimensional Bin Packing problem provides a framework for considering this type of problem. Given a list L of items,

$$L=\{a_1, a_2, ..., a_n\}_{[1]},$$

each with a size in the range (0,1], the problem seeks the minimum number of unit capacity bins the items can packed into, while obeying the capacity constraints.

The Subset Sum Algorithm

There are many heuristics used to solve this problem. One approach is to take one bin, pack it as full as possible and then close it. This process then continues until no items are left to pack[2]. This approach is termed the Subset Sum algorithm.

This heuristic however, does not perform well in the worst case. Consider the input shown in Figure 1 to the right, representative of the worst case. The Subset Sum algorithm uses 10 bins to pack all items (Figure 2), compared to 7 bins in the optimal packing (Figure 3).

Variants of the Subset Sum Algorithm

Two variants of the Subset Sum Algorithm for Bin Packing were proposed by Caprara and Pferschy[3].

The first, the Largest Items First Subset Sum (LSS), first opens bins for all large items (size greater than one half). The remaining space in these bins is packed using the usual Subset Sum algorithm. The items remaining after all of these bins are used are packed as usual.

The second, Largest Remaining Item First Subset Sum (LRSS), packs the largest item remaining in the unpacked items as each bin is opened, with the remaining space being packed with Subset Sum.

These approaches perform better than the Subset Sum algorithm, as shown in Figure 4.

Asymptotic Worst Case Ratio

The asymptotic worst case ratio of an algorithm is a measurement of how badly an algorithm may perform in the worst case. It describes the maximum ratio of the number of bins used by the algorithm to that used in the optimal solution, as the number of bins used in the optimal solution approaches infinity.

The asymptotic worst case ratio for LSS is known to be between 1.37 and 1.44[3][4].

For LRSS, the bounds are 1.30 and 1.44[3].

Project Aim

The aim is to determine tight bounds on the asymptotic worst case ratio on the two variants of the Subset Sum algorithm for Bin Packing.
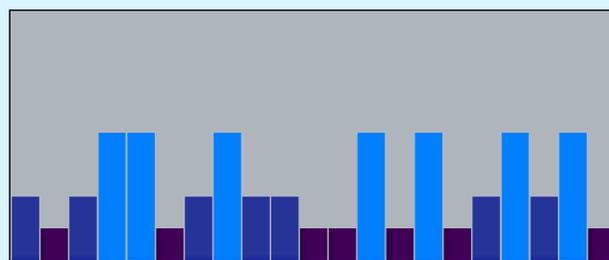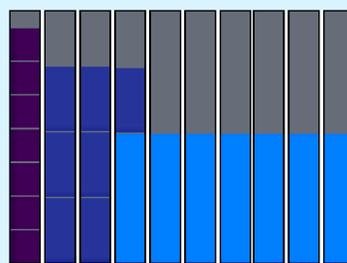

Figure 1:
Example of Bad Input for Subset Sum

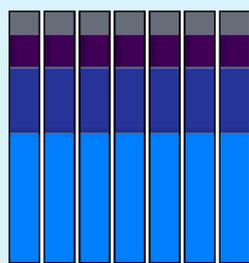
Figure 2:
Packing produced by Subset Sum for input

Figure 3:
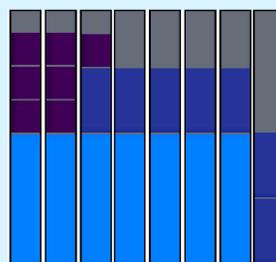Optimal packing for input

Figure 4:
Packing produced by LSS/LRSS for input

SELFISH ITEMS

Game Theory

We can also consider the Bin Packing problem as a game played by selfish items, each of whom want to minimize their packing cost.

Weighting Function

To use this approach to determine the asymptotic worst case ratio of LSS and LRSS we must define a weighting function that gives this ratio for the worst bin in the packing[4]. To find this bin, we first charge costs to items in the bins packed by the algorithm. These costs are then assigned to the bins in the optimal packing that contains those items. This process is shown in Figure 5 below. The asymptotic worst case ratio is then given by the highest cost bin in the optimal packing.
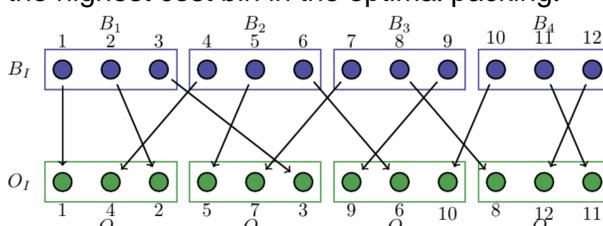

Figure 5: Weightings charged to items

The weighting functions used to charge items are based on the processes that LSS and LRSS use to pack items.

LARGEST ITEMS FIRST

Weighting Function

The different weighting cases are as follows:

- Large item: 1 if no other items are packed in the bin, the size of the bin otherwise

- Small items when bin includes a large item: a proportion of the space remaining

- No large items in the bin: a proportion of the size of the bin if the bin is full enough; the size of the item otherwise

- If the bin is less than two thirds full: a proportion of two thirds based on the size of the item

Result

The worst case input that includes a large item is the same as for the Subset Sum. However, as large items must be packed optimally, the ratio is 1.202.

The worst case input that does not include a large item is known[4] and has a ratio of 1.376. In this packing, the first item is just over one third, and then each further item uses just over half of the space remaining in the bin.

LARGEST REMAINING ITEM FIRST

Weighting Function

The weighting function is similar to that for Largest Items First. However, the large item check is replaced for a largest item check, and thus there are no cases for when no large items are included.

Result

The worst case input is the same as for LSS. However, the asymptotic worst case ratio is 1.303. This improvement compared to LSS is due to the largest items being packed first into bins, and thus being packed optimally.

CONCLUSION

Our work has determined tight bounds on two variants of the Subset Sum algorithm for Bin Packing. We found that these were the previously known lower bounds.

We found that the worst case for Largest Items First includes no large items. Instead the global worst case is equal to the previously known worst case in the absence of large items, with an asymptotic worst case ratio of 1.377.

We found the same worst case input for Largest Remaining Item First. However, it has a better asymptotic worst case ratio of 1.303.

[1] D.S. Johnson, "Near-Optimal Bin Packing Algorithms", Ph.D. dissertation, MIT, 1973.

[2] R.L. Graham, "Bounds on multiprocessing anomalies and related packing algorithms", in *AFIPS Conference Proceedings*, 1972, pp. 205-217.

[3] A. Caprara and U. Pferschy, "Modified subset sum heuristics for bin packing", *Information Processing Letters,* vol. 96, no. 1, pp. 18-23, 2005.

[4] L. Epstein, E. Kleiman and J. Mestre, "Parametric Packing of Selfish Items and the Subset Sum Algorithm", *WINE 2009*, vol. 5929, pp.67-78, 2009.