CSE 642: Scribe Notes
April 4, 2025

Russell Bentley

Friday 4th April, 2025

1 Introduction

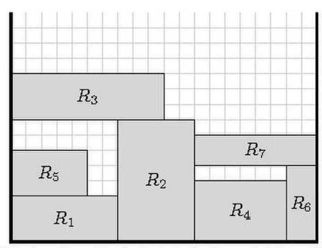
Today Rezaul continued to lead the discussion on their generalization of the strip-packing problem to include multiple colors. Today's discussion focussed on the issue of approximating solutions.

The motivation for studying this problem is scheduling the runtime for p programs using a limited amount of shared memory, M. We assume that each program is specified as a set of blocks with a fixed width (memory requirement) and height (runtime requirement). Additionally, we have the constraint that each program cannot use more than αM memory at a given point in time where $\alpha \in [1/p, 1)$. The goal is optimize the completion time for all programs.

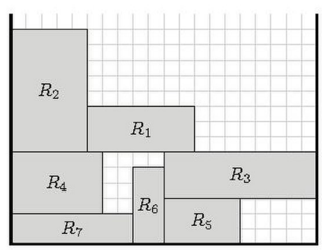
In the case where $p = 1$, this reduces to the strip-packing problem, see the following example from wikipedia.

Input Rectangles

i	1	2	3	4	5	6	7
$w(i)$	7	5	10	6	5	2	8
$h(i)$	3	8	3	4	3	5	2



BL for $L = (1, 2, 3, 4, 5, 6, 7)$ and $W = 20$



BL for $L = (7, 6, 5, 4, 3, 2, 1)$ and $W = 20$

Rezaul has collected some related papers into a google drive.

2 Multi-color Strip Packing

The hardness of this problem follows from strip-packing, when $p = 1$, and in turn bin packing which is NP-Hard. Last week Lucas showed how to prove the hardness using partitions.

Also, last week we started working on an approximation algorithm We'll consider two special cases.

1. There are only 2 programs
2. There are p programs but each includes one task or program

We can prove hardness since when $p = 1$, we have basic strip packing problem which has been proven to be hard. Last week we considered special case 1.

Lucas said if we had k colors, its trivial to have a k-times best approximation. The idea is to use a bin-packing approximation for one program at a time, restricted to αM memory, and then stack their solutions. If the bin-packing approximation has an approximation factor of A, then we would have a $kA + \epsilon$ approxmation for the multi-color strip-packing problem.

- Mayank pointed out this analysis is for completion opt, might be stronger for makespan opt?
- Rezaul pointed out, why not bin pack all rectangles into αM space?

There was some discussion about the exact value of this approximation. If the first program has AOpt, second is $2A$Opt, we would get $\frac{k(k+1)}{2}A$Opt in total?

How to do better:

- What about shelfing?
- Sorting rectangles some how?

Without the M constraint, we can sort rectangles by completion time and doing smaller jobs first will yield optimal completion time. I.e. reduce to $1D$ case.

Another angle discussed was what if all rectangles have the same width? When all widths are the same, but not equal to M, we could sort the rectangles in non-decreasing order of height Start arranging left to right and bottom to top. Undecided if this would be optimial, but it is a natural generalization of the $1D$ problem.