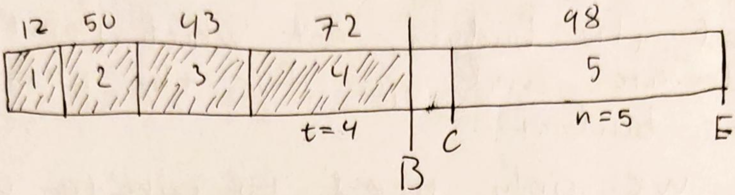
Task 2 a) greedy cheapest first.

reminds me of OS-scheduling: SJF.

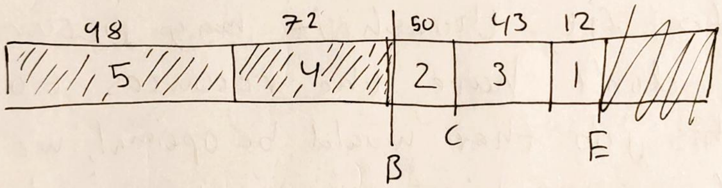


With n projects, assume t is the last one. Resources required to complete t is C, such that B ≤ C. To complete all projects, E.

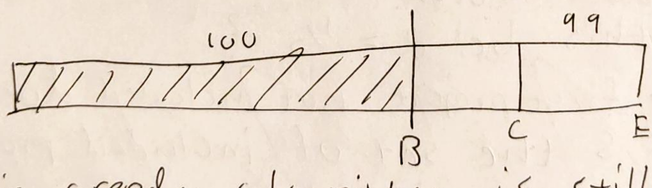
If all projects had a uniform profit value, this would be optimal, but we are disregarding the profits completely now, thus we are always less than or equal to the optimum.

Because of this, I posit that there is no constant approximation factor.

b)



Consider:

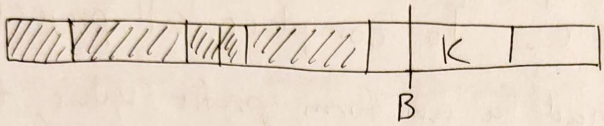


This greedy algorithm is still not optimal, since it disregards project costs.
Thus I still don't think we are quite able to show any approximation factors.

c) We have to consider both profit and cost, so sort decreasingly by density p_i/r_i. Then all resources spent will be spent optimally. Sorting is polynomial time, so the greedy algorithm is too.

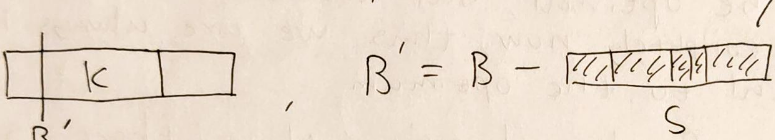
d) With $x_i \in \{0, 1\}$, we can still attempt the algorithm from c). We still take the highest density jobs until we meet one that would exceed the budget, at which point we must deviate and opt for the next project that fits our budget.

I'm saying we only need to consider what happens from after a problematic project shows up:



$\cancel{\sum_{i \in S} p_i}$

Can be seen as / reduced to



However in this case, we can't find another project to fit, though it may occur.
Since we don't have the resources to complete this job that would be optimal, we have to settle for something worse, thus inducing an approximation factor.
Why should this be $\alpha = 1/2$?

~~Let k be the first project not included due to budget violation, S the set of included projects. The optimum can either include k or not~~

1. if $p_k \geq \sum_{i \in S} p_i$

It shouldn't

The approximation is almost correct; we need to account for a case like

$p_1 = 1 \quad r_1 = 1$, ~~p_2~~, ~~p_m~~

~~$p_2 = 0.95$~~ $p_2 = 1.95 \; r_2 = 2$, $\quad B = 2$

then sorting by p_i / r_i yields project 1 first, but then we don't have funds for 2, which should've been picked instead since it is optimum.

Thus, modify the algorithm to take

$$\max\left(\sum_{i \in S} p_i \,,\; p_k\right),$$

where S is selected projects before skipping and k is the first skipped project.

This achieves an $\alpha = 1/2$ ~~due to 2 cases.~~

We have $OPT \leq OPT_{frac} \leq \sum_{i \in S} p_i + p_k$, where OPT is this binary problem and OPT_{frac} is a)–c) where we allowed fractional solutions. These optima are bounded by

$\sum_{i \in S} p_i + p_k > B$, since these are the greedy highest profit projects. but

$\sum_{i \in S} p_i \geq 0$ and $p_k \geq 0$, so

$$\sum_{i \in S} p_i + p_k \leq 2 \cdot \underbrace{\max\left(\sum_{i \in S} p_i \,,\; p_k\right)}_{APX}$$

$$\Rightarrow \; OPT \leq 2 \cdot APX \;\Rightarrow\; \alpha = 1/2$$

■