[There is no final exam!!]

Oct & Nov:

- Spend some time on ILP algorithms, related optimization.
- Weekly homeworks (light) to illustrate ↺
- We'll start to have presentations on proposals
- Long presentations in November

=

- Today! Bin packing, 2 bins:

n items, sizes $a_1, \ldots, a_n > 0$

say $a_1 + \ldots + a_n = 2$ ideally we could split $a_1, \ldots, a_n$ into two groups, each adding to one.

Problem: ILP relax to LP give very bad bounds...

Problem: Bin packing, or partition (2 bins) is NP complete

- Today: Bin Packing:

W Fernandez de la Vega & G.S. Lueker. "Bin packing can be solved within $1+\varepsilon$ in linear time." Combinatorica (1) 349-355, 1981.

- Upshot: You can have Gurobi (or some other optimizer) give an approximate bin packing solution. You could even program this by hand without any LP solver...

We have $a_1, \ldots, a_n$, sum $= 2$ want divide into two sets of roughly equal size.

We had a wonderful illustration with chalk...

First: $a_1 \geq a_2 \geq \dots \geq a_n \geq 0$

Imagine that we want to get with 10% of optimal. Say

$a_1 \geq \dots \geq a_k \geq .1$

$a_{k+1}, \dots, a_n < .1$

(1) Look at all ways of splitting up $a_1, \dots, a_k$

(in practice, you'd Branch and Bound over $a_1, a_2, \dots$ in that order)

Remark: since each of $a_1, \dots, a_k$ is $\geq .1$,

$k \leq 20$ (since $a_1 + \dots + a_k \leq 2$)

Worst case: for each $M \subset \{1, \dots, k\}$

$$\max\left(\sum_{m \in M} a_m \, , \sum_{\substack{m \notin M \\ m \in \{1,\dots,k\}}} a_m \right)$$

so M subset of $\{1, \dots, k\}$, at largest $\{1, \dots, 20\}$

$\# M\text{'s} \leq 2^{20}$

Two cases:

If $a_1, \ldots, a_k$ have best partition

$\{1, \ldots, k\}$ into m, m',

$$\min_{m,m'} \left(\max \left(\sum_{m \in m} a_m, \sum_{m \in m'} a_m \right) \right) = \begin{Bmatrix} 1.315 \\ [illegible] \end{Bmatrix} \text{ or } \geq 1$$

i.e. dividing some (the large elements) can't be done with one bin at least 1.315

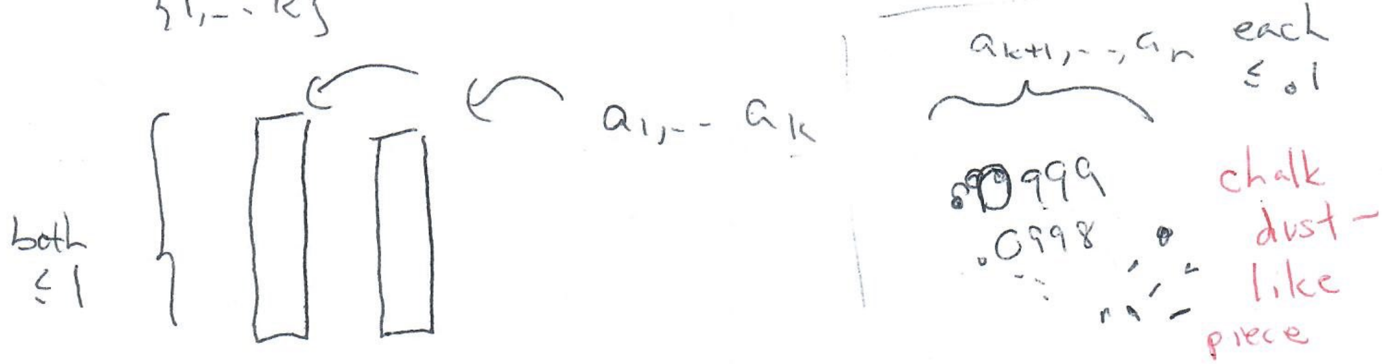
Then add all $a_{k+1}, \ldots, a_n$ to smaller part and get

1.315 versus $2 - 1.315 = 0.685$.

Then you've found the best packing...

If

$$\min_{\substack{m, m' \text{ partition} \\ \{1, \ldots, k\}}} \max \left(\sum_{m \in m} a_m, \sum_{m \notin m} a_m \right) < 1$$



Then back and forth with $a_{k+1}, \ldots, a_n$, and get with a difference of .1 : worst .95 one bin, 1.05 other bin.

Best possible: 1 in one bin, 1 in other bin, so now within .05 optm