MATH 567: Lecture 22 (04/01/2025)

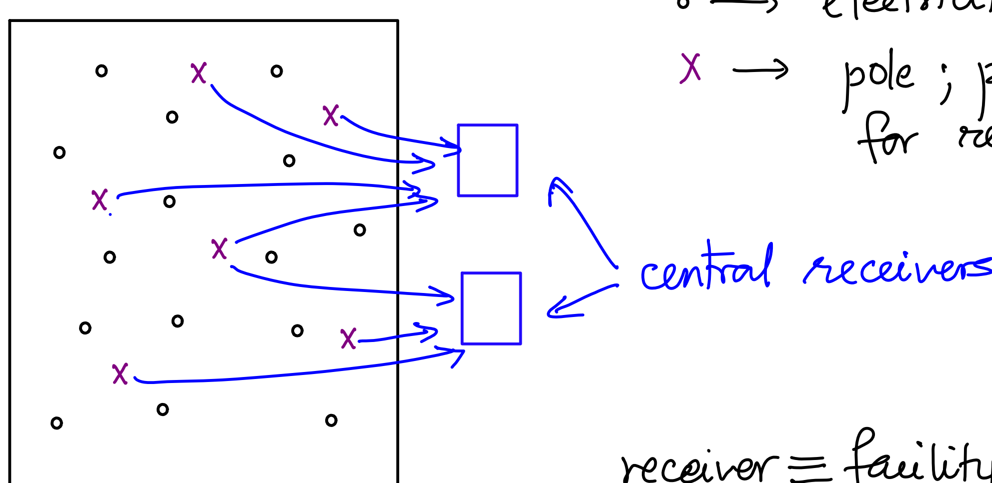
Today: * set covering problem (heuristics)

Solving Large sized Integer Programs: Set Covering Problem

Given:
1) n customer locations
2) m candidate facility locations
3) for each candidate facility location, the subset of customers that can be covered.

Assume no limits on # customers a facility can serve — hence we could look at it as an instance of uncapacitated facility location (UFL) problem.

Application: Receiver location problem for reading electricity/gas meters.



o → electricity meter

x → pole; potential locations for receivers/amplifier

receiver ≡ facility   meter ≡ customer

The meters transmit readings to (at least one) receiver, which amplifies it before transmitting to a central receiver.

Goal: Identify which poles to locate receivers on, so that we minimize the total # recievers (i.e., facilities) used such that every meter (i.e., customer) can transmit to at least one receiver.

Such problems are often quite big, and hence cannot be handled easily as (M)IPs. We consider heuristics.
→ algorithms that are not guaranteed to find the optimal solution.

Also, we do not get any measures of the quality of the solutions found.

But, they often work well in practice!

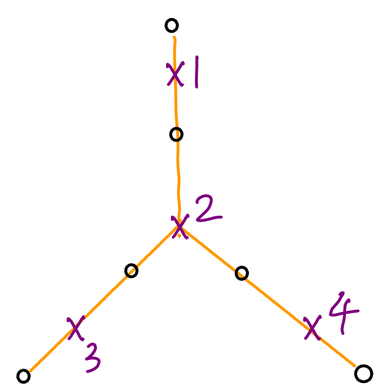
Heuristics

1. Greedy algorithm: In each step, pick the pole that covers the **largest number of uncovered meters**.

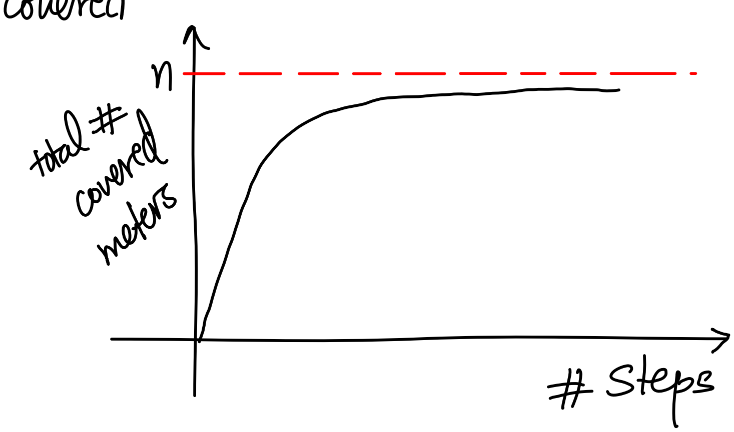
   Break ties arbitrarily.

   In general, will not give optimal solution.

   Here, greedy gives $\{2,1,3,4\}$, while optimal solution is $\{1,3,4\}$.



As the heuristic runs, the #-covered meters "plateaus" out.

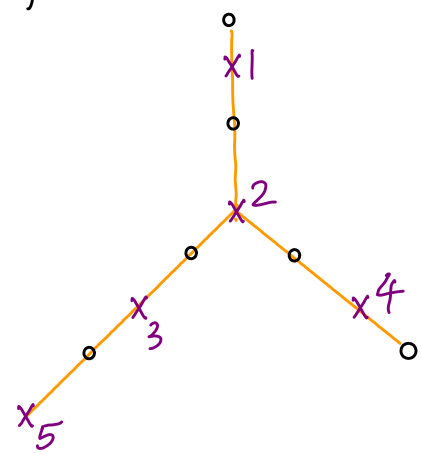


Cleaning up the solution

If removing the i^{th} pole from the set of selected poles leaves all meters covered (as covered up to now), then remove that pole. Repeat for $i=1,\ldots,p$ after p-steps, for all p (or, say, repeat after every 10^{th} pole).

Clean up gives optimal solution in the previous example.

But in this example, if greedy gives $\{2,1,4,5\}$, clean up will do nothing.

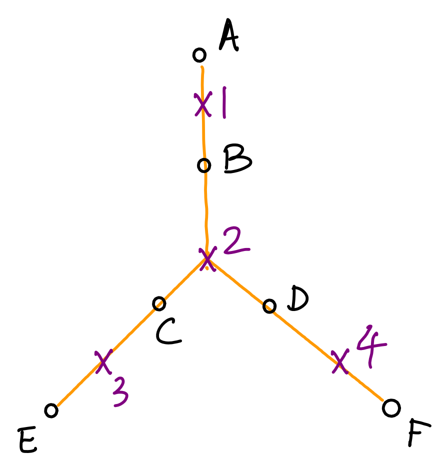


(2) **Modified Greedy algorithm** (Balas and Ho)

- Uses more foresight than greedy
- IDEA: Define a scoring function, and in each step, pick the pole with the largest value.

Def A meter is called **hard to cover** if the number of poles that cover it is minimal.

Here, A, E, F are hard-to-cover.



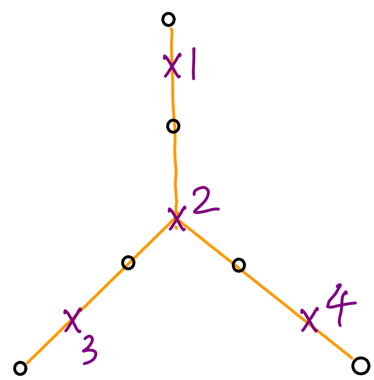
For pole j, define

$$\text{Score}_1(j) = \begin{cases} \text{total \# meters covered by pole } j \textbf{ if } \text{it covers at least one hard-to-cover meter} \\ 0, \quad \text{otherwise.} \end{cases}$$

and

$$\text{Score}_2(j) = (\text{\# meters covered by pole } j) \times (\text{\# hard-to-cover meters covered by pole } j).$$

Modified greedy works in this example. $\text{Score}_1(2) = 0$ and $\text{Score}_1(j) = 2$ for $j = 1,3,4$, at start, and do not change as the algorithm proceeds. Hence, we select $\{1,3,4\}$.



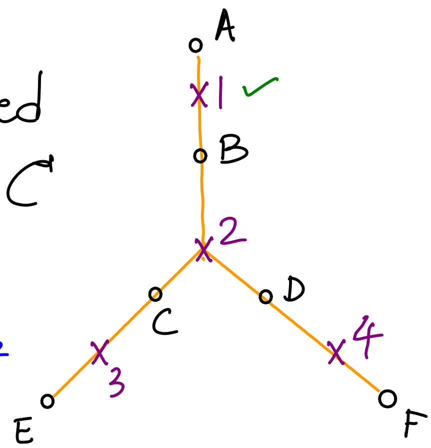
A Generalized Scoring Function

Def Suppose modified greedy at some intermediate step has selected a subset $P' \subseteq P$ of the poles. Let $m_{P'}$ be the minimum # poles that can cover a meter (not yet covered). For $t \in \mathbb{Z}_{>0}$, an uncovered meter is **t-hard to cover** if the # poles that can cover it is $< m_{P'} + t$.

So, 1-hard to cover $(t=1) \equiv$ hard-to-cover as previously defined.

Example Let $P' = \{1\}$. Then $m_{p'} = 1$ (for E and F). Meter C is covered by both poles 2 and 3. Hence, C is 2-hard to cover, as $\underset{\text{poles } 2,3}{2} < \underset{m_{p'}}{1} + \underset{t=2}{2}$.

$t=2$



E, F are 1-hard to cover ($t=1$).

C, D are <u>also</u> t-hard to cover for all $t \geq 3$.

Let $s(j,t) = \#$ t-hard to cover meters covered by pole j.

We define

$$\text{Score}_g(j) = s(j,\infty) \prod_{t=1}^{k} [s(j,t)]^{1/t}$$

general

$1/t$ weighs 1-hard-to-cover meters > 2-hard-to-cover meters > 3-hard-to-cover ⋮

$s(j,\infty)$ → total # uncovered meters covered by j

where $k \geq 1$ is a fixed positive integer.

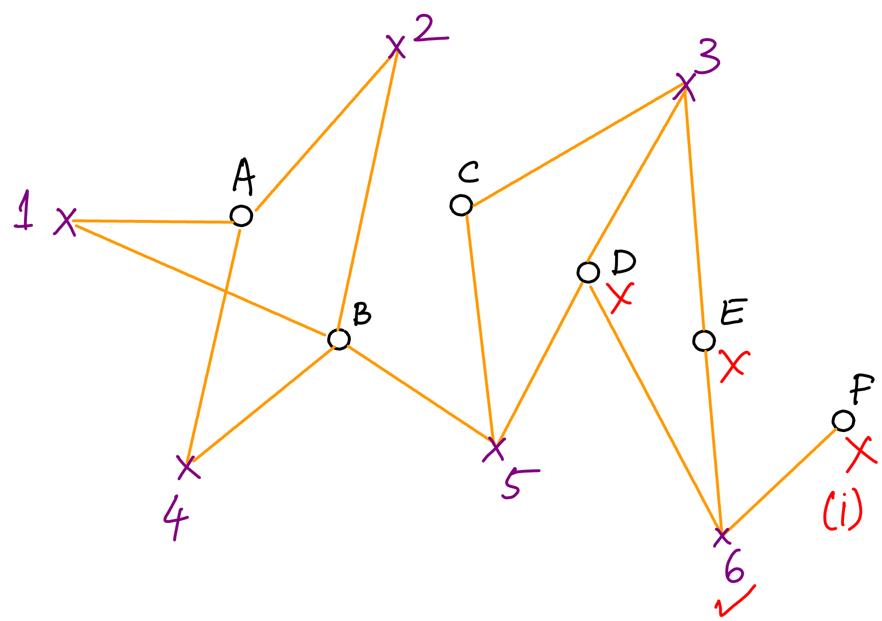
Note

1. $\text{Score}_g(j) = \text{Score}_2(j)$ when $k=1$.
2. $\text{Score}_g(j) = \text{Score}_1(j)$ if there is a unique meter covered by pole j that is 1-hard to cover
3. higher $k \Rightarrow$ more foresight

(3) Preprocessing

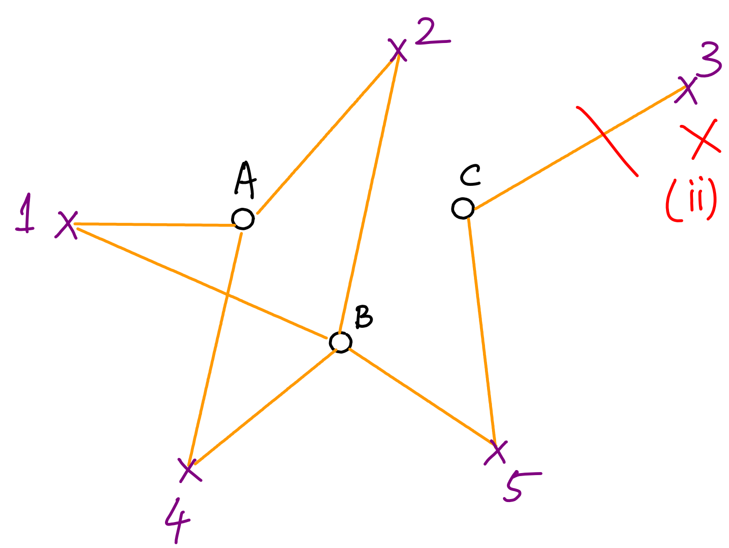
Reduces the size of the problem, but does not typically give an optimal solution (except in trivial cases).

We illustrate the main steps on an example.

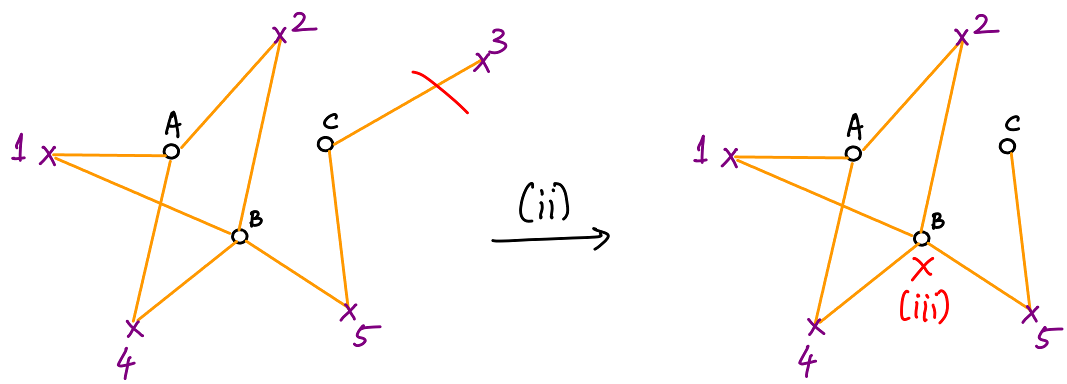


(i) Only 6 covers F $\Longrightarrow$ Choose 6, delete D, E, F

(as pole 6 covers D, E, F).

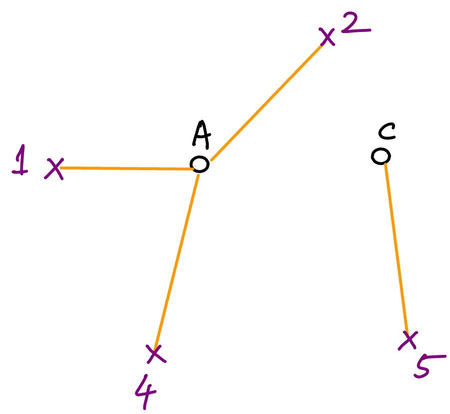


(ii) Now pole 5 covers all meters that pole 3 covers $\Rightarrow$ delete pole 3.



(iii) $\{1,2,4,5\}$ and $\{A,B,C\}$ are left. Now,

poles covering A also cover B $\Longrightarrow$ delete B.



another application of step(i) here!

(iv) Left with $\{1,2,4,5\}$ and $\{A,C\}$.
Optimal solution: pick 5 (as only 5 covers C), and pick one out of 1,2,4, say, 1 $\Longrightarrow$ $\{1,5\}$.
(On larger instances, we run greedy/modified greedy on this smaller instance).

(v) Extend (optimal) solution in Step (iv) to an (optimal) solution to the whole problem by adding pole 6 chosen in Step (i) $\Longrightarrow$ Solution is $\{1,5,6\}$.