MATH 567 – Lecture 23 (04/09/2015)

Solving large sized integer programs – Set covering problem

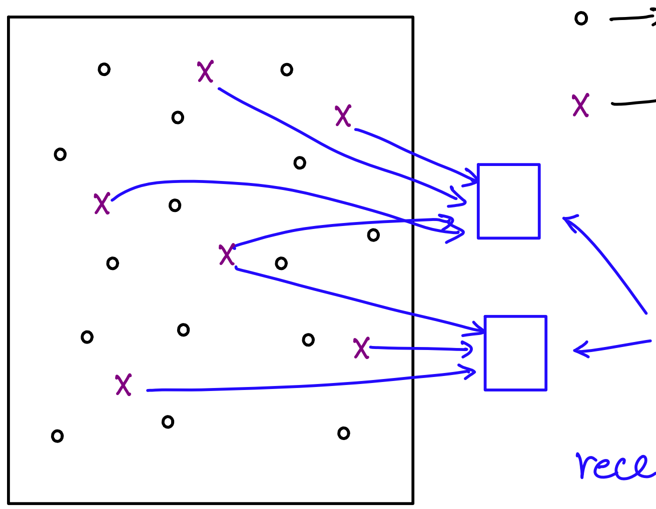
also called the facility location problem

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

incorporates any distance information, e.g., coverage radius.

Assume there is no limit on how many customers each facility can serve – hence we look at the uncapacitated facility location (UFL) problem.

Application: Receiver location problem



o → electricity meters

x → poles; potential locations for receivers (amplifiers)

central receivers

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) could transmit to at least one receiver.

Such problems are often quite big, and hence cannot be handled easily as MIPs. We consider heuristics.

↳ algorithms that are not guaranteed to find the optimal solution.

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

But, they often work well in practice!

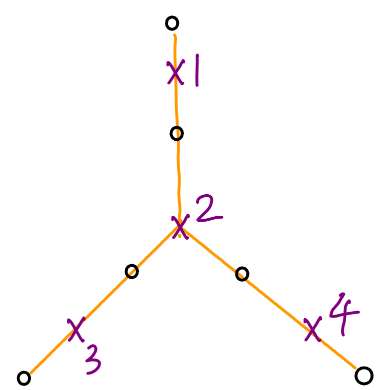
Heuristics

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

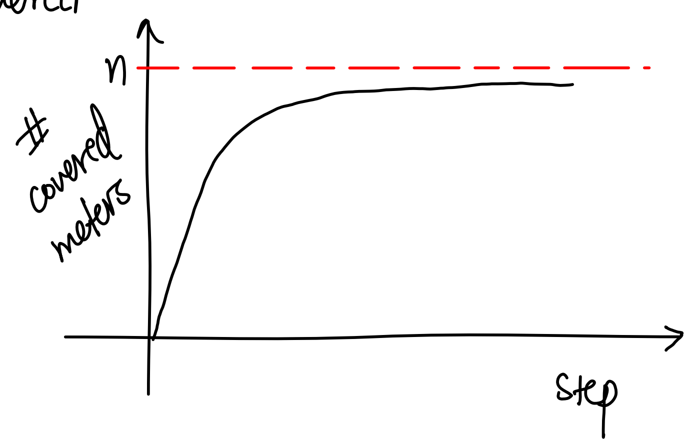
   Break ties arbitrarily.

   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.

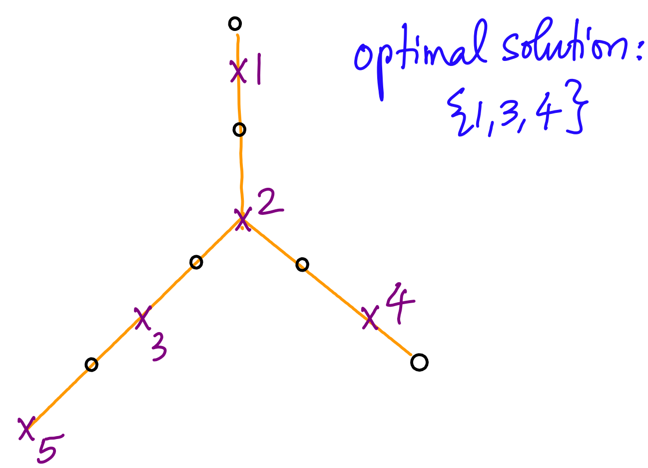


Cleaning up the solution

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 $i = 1, 2, \ldots, p$ after p-steps, for all p (or, say, repeat after every 10^{th} step).

Clean up gives the correct solution in the previous example.

But in this example, if Greedy gives $\{2, 1, 4, 5\}$, cleaning up does not change anything.

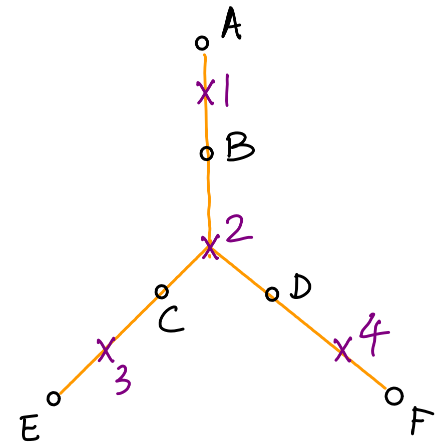


(2) **Modified Greedy algorithm**

- Uses more foresight than Greedy
- proposed by Balas and Ho
- IDEA: Define a scoring function, and in each step, pick the pole with the largest value of the scoring function.

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

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



For pole j, define

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

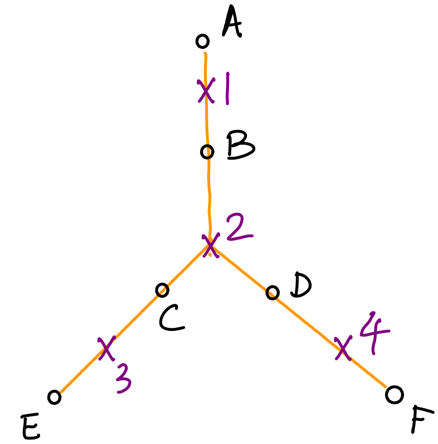
and

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

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

A Generalized Scoring function

Def Suppose modified greedy has selected a subset P of the poles. Let m_P be the minimum number of 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 $< m_P + t$.



Example Let $P = \{1\}$. $m_P = 1$ now.

The # poles that can cover C is 2 (poles 2 & 3). Hence C is 2-hard to cover (as $2 < 1 + \underset{t}{2}$).

C is also t-hard to cover for $t \geq 3$.

Notice that 1-hard-to-cover ($t=1$) is the same as hard-to-cover as defined previously.

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

We define

weighs 1-hard-to-cover meters
> 2-hard-to-cover meters
> 3-hard-to-cover meters
⋮

$$\mathrm{Score}_g(j) = \underbrace{s(j,\infty)}_{\#\text{ uncovered meters}} \cdot \prod_{t=1}^{k} [s(j,t)]^{1/t}$$

"general"

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

Notes

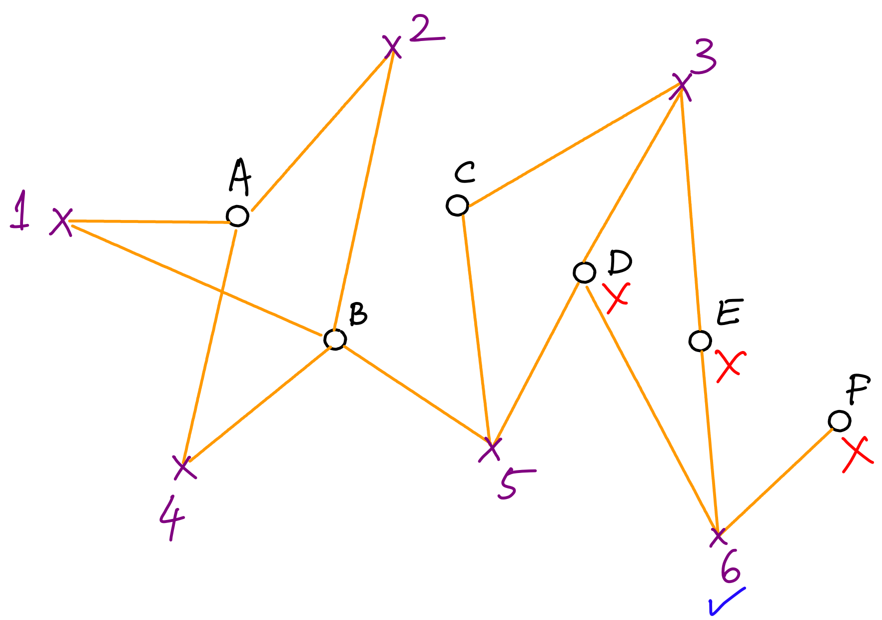
(i) $\mathrm{Score}_g(j) \equiv \mathrm{Score}_1(j)$ if there is a unique uncovered meter that is 1-hard to cover.

(ii) higher $k \Rightarrow$ more foresight.

(3) **Preprocessing**

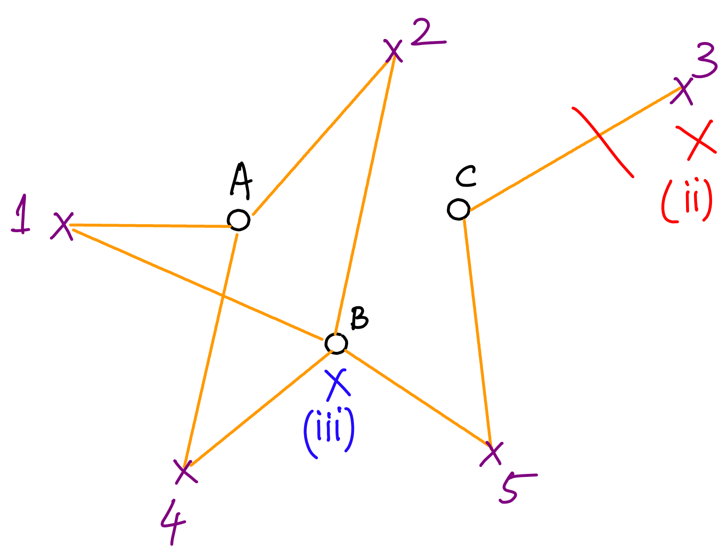
Reduces the size of the problem, but does not typically give an optimal solution (except in trivial cases).

We illustrate the main steps on an example.



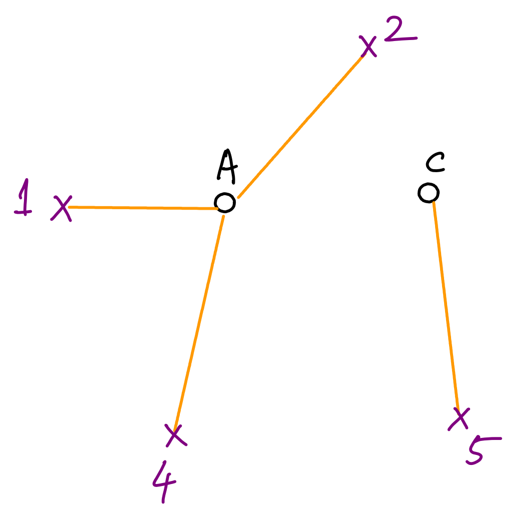
(i) Only 6 covers F $\Longrightarrow$ choose 6, delete D, E, F

(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. $\Rightarrow$ 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 $\Rightarrow$ $\{1,5\}$.

(v) Extend optimal solution in Step (iv) to the optimal solution of the original problem by adding $\{6\}$ chosen in Step (i).
$\Rightarrow$ Optimal solution $=\{1,5,6\}$.

In the next lecture, we will talk about how to implement the preprocessing steps on the binary incidence matrix (of the poles and meters).