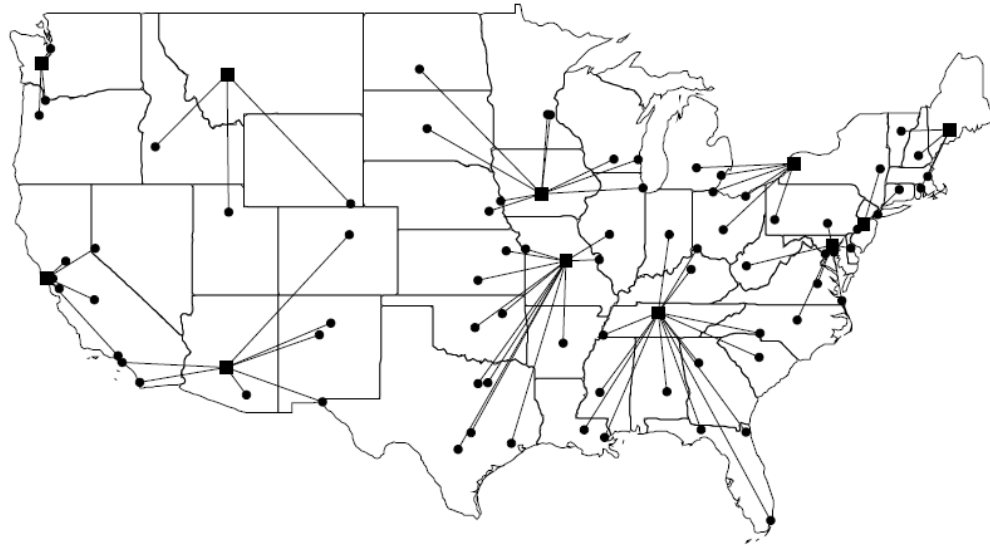


# LEC012 Facility Location I

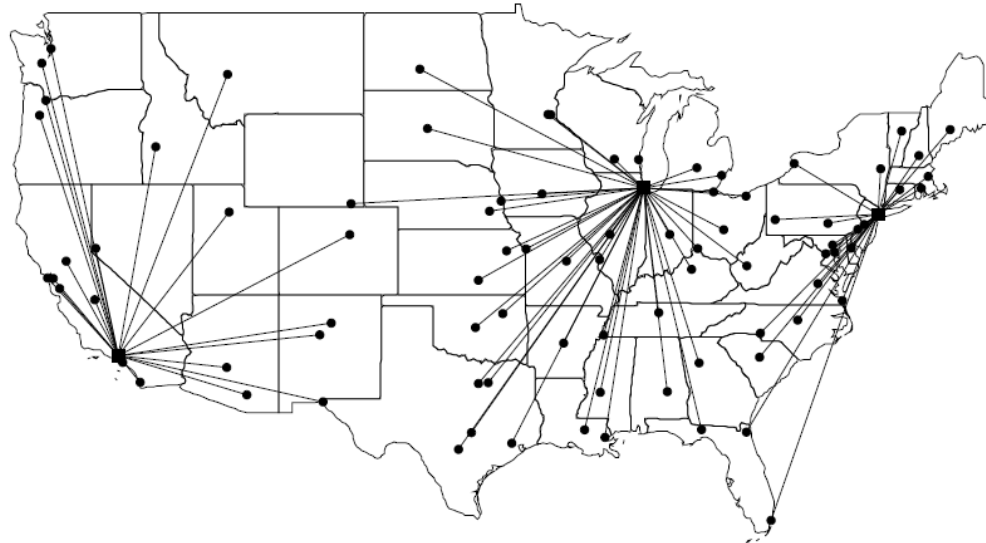
VG441 SS2020

Cong Shi  
Industrial & Operations Engineering  
University of Michigan

# Configurations



(a) Many facilities open.



(b) Few facilities open.

# Formulation

- Sets

$I$  = set of customers

$J$  = set of potential facility locations

- Parameters

$h_i$  = annual demand of customer  $i \in I$

$c_{ij}$  = cost to transport one unit of demand from facility  $j \in J$  to customer  $i \in I$

$f_j$  = fixed annual cost to open a facility at site  $j \in J$

- Decision Variables

$x_j$  = 1 if facility  $j$  is opened, 0 otherwise

$y_{ij}$  = the fraction of customer  $i$  's demand that is served by facility  $j$

# Transportation costs $c_{ij}$

- Euclidean distance

$$\sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2}$$

- Manhattan or rectilinear metric

$$|a_1 - a_2| + |b_1 - b_2|$$

- Great circle

$$r \arccos(\sin \alpha_1 \sin \alpha_2 + \cos \alpha_1 \cos \alpha_2 \cos(\Delta\beta))$$

- GPS/MAP distance

# MILP Formulation

Decision variables:

$x_j = 1$  if facility  $j$  is opened, 0 otherwise

$y_{ij}$  = the fraction of customer  $i$  's demand that is served by facility  $j$

$$\begin{array}{ll}\text{minimize} & \sum_{j \in J} f_j x_j + \sum_{i \in I} \sum_{j \in J} h_i c_{ij} y_{ij} \\ \text{subject to} & \sum_{j \in J} y_{ij} = 1, \quad \forall i \in I \quad \longrightarrow \text{Assignment Constraint} \\ & y_{ij} \leq x_j, \quad \forall i \in I, \forall j \in J \quad \longrightarrow \text{Linking Constraint} \\ & x_j \in \{0, 1\}, \quad \forall j \in J \\ & y_{ij} \geq 0, \quad \forall i \in I, \forall j \in J\end{array}$$

# Python + Gurobi Time!



**GUROBI**  
OPTIMIZATION

# Lagrangian Relaxation

- Introduce a penalty term:  $\sum_{i \in I} \lambda_i \left( 1 - \sum_{j \in J} y_{ij} \right)$

$$\begin{aligned} \text{minimize} \quad & \sum_{j \in J} f_j x_j + \sum_{i \in I} \sum_{j \in J} h_i c_{ij} y_{ij} + \sum_{i \in I} \lambda_i \left( 1 - \sum_{j \in J} y_{ij} \right) \\ & = \sum_{j \in J} f_j x_j + \sum_{i \in I} \sum_{j \in J} (h_i c_{ij} - \lambda_i) y_{ij} + \sum_{i \in I} \lambda_i \\ \text{subject to} \quad & y_{ij} \leq x_j, \quad \forall i \in I, \forall j \in J \\ & x_j \in \{0, 1\}, \quad \forall j \in J \\ & y_{ij} \geq 0, \quad \forall i \in I, \forall j \in J \end{aligned}$$

# Lagrangian Relaxation

- It turns out that this relaxation is easy to solve:

$$\begin{array}{ll}\text{minimize} & = \sum_{j \in J} f_j x_j + \sum_{i \in I} \sum_{j \in J} (h_i c_{ij} - \lambda_i) y_{ij} + \sum_{i \in I} \lambda_i \\ \text{subject to} & y_{ij} \leq x_j, \quad \forall i \in I, \forall j \in J \\ & x_j \in \{0, 1\}, \quad \forall j \in J \\ & y_{ij} \geq 0, \quad \forall i \in I, \forall j \in J\end{array}$$

- Observe that if we open facility  $j$ , i.e., set  $x_j = 1$

$$\beta_j = \sum_{i \in I} \min \{0, h_i c_{ij} - \lambda_i\} \quad \longrightarrow \quad \text{"benefit"}$$

- So we should open  $j$  if and only if  $\beta_j + f_j < 0$



# Lagrangian Relaxation

$$\begin{array}{ll}\text{minimize} & = \sum_{j \in J} f_j x_j + \sum_{i \in I} \sum_{j \in J} (h_i c_{ij} - \lambda_i) y_{ij} + \sum_{i \in I} \lambda_i \\ \text{subject to} & y_{ij} \leq x_j, \quad \forall i \in I, \forall j \in J \\ & x_j \in \{0, 1\}, \quad \forall j \in J \\ & y_{ij} \geq 0, \quad \forall i \in I, \forall j \in J\end{array}$$

Solution:

$$\begin{aligned}\bar{x}_j &= \begin{cases} 1, & \text{if } \beta_j + f_j < 0 \\ 0, & \text{otherwise} \end{cases} \\ \bar{y}_{ij} &= \begin{cases} 1, & \text{if } \bar{x}_j = 1 \text{ and } h_i c_{ij} - \lambda_i < 0 \\ 0, & \text{otherwise} \end{cases}\end{aligned}$$

Objective value:

$$z_{\text{LR}}(\lambda) = \sum_{j \in J} \min \{0, \beta_j + f_j\} + \sum_{i \in I} \lambda_i$$

# Now Construct UB

- Use the “opened” facility in the relaxed solution
- Assign each customer to its nearest open facility

# Updating Multipliers

- Adjust the penalty multipliers:

If  $\sum_{j \in J} y_{ij} = 0$ , then  $\lambda_i$  is too small; it should be increased.

If  $\sum_{j \in J} y_{ij} > 1$ , then  $\lambda_i$  is too large; it should be decreased.

If  $\sum_{j \in J} y_{ij} = 1$ , then  $\lambda_i$  is just right; it should not be changed.

- Then a natural updating rule becomes

$$\lambda_i^{t+1} = \lambda_i^t + \Delta^t \left( 1 - \sum_{j \in J} y_{ij} \right) \quad \text{where the step-size is}$$

$$\Delta^t = \frac{(\text{UB} - z_{\text{LR}}(\lambda^t))}{\sum_{i \in I} \left( 1 - \sum_{j \in J} y_{ij} \right)^2}$$