Modelamiento y Optimización

Clase 3

Gonzalo Muñoz

18 de Marzo 2024

Clase pasada: Localización (Facility location, P-Centering)

Queremos instalar P nuevos almacenes en una comuna. Consideramos J posibles ubicaciones, y tenemos un conjunto K de almacenes ya existentes. Además, tenemos un conjunto I de clientes que utilizarían estos almacenes. El cliente $i \in I$ está a distancia d_{ij} del lugar $j \in J \cup K$.

Queremos minimizar la distancia máxima entre un almacén y un cliente.

$$\min \quad t$$

$$\sum_{j \in J \cup K} d_{ij} x_{ij} \leq t \qquad \forall i \in I$$

$$\sum_{j \in J} y_j = P \qquad \text{(almacenes a instalar)}$$

$$\sum_{j \in J \cup K} x_{ij} = 1 \qquad \forall i \in I \quad \text{(toda persona tiene un almacén)}$$

$$\text{(si } i \text{ se asigna a } j\text{, } j \text{ debe existir)} \qquad x_{ij} \leq y_j \qquad \forall i \in I \ \forall j \in J$$

$$x_{ij} \in \{0,1\}$$

$$y_j \in \{0,1\}$$

Localización (Variantes)

Ahora suponer que un almacén sólo puede atender a clientes que están a una distancia D o menos. Formule un problema lineal para determinar cómo cubrir a la mayor cantidad de clientes abriendo un número máximo de almacenes P.

$$\max \sum_{i \in I} (1 - u_i) \qquad u_i = \begin{cases} 0 & \text{si } i \text{ es asignado} \\ 1 & \sim \end{cases}$$

$$\sum_{j \in J} y_j \leq P$$

$$\sum_{j \in J \cup K} x_{ij} + u_i = 1 \qquad \forall i \in I$$

(nueva) $$x_{ij} = 0 \qquad \forall i,j \; d_{ij} > D$$

✓ $$x_{ij} \leq y_j \qquad \forall i \in I \; \forall j \in J$$

$$x_{ij} \in \{0,1\}$$
$$y_j \in \{0,1\}$$
$$u_i \in \{0,1\}$$

Asignación (Matching)

Supongamos que tenemos 3 máquinas y 3 tareas. Deseamos asignar una tarea a cada máquina, de manera de minimizar la suma de los tiempos que están ocupadas las máquinas. El tiempo (minutos) que toma cada tarea en cada máquina está representado en la siguiente tabla:

	Tarea 1	Tarea 2	Tarea 3
Máquina 1	3	4	5
Máquina 2	3	2	4
Máquina 3	7	6	9

¿Cómo modelar esta asignación como un problema de optimización?

Variables :

$$x_{ij} = \begin{cases} 1 & \text{si máquina } i \text{ se asigna a tarea } j \\ 0 & \sim \end{cases}$$

Asignación (Matching)

$$\min \quad 3x_{11} + 4x_{12} + 5x_{13} + 3x_{21} + 2x_{22} + 4x_{23} + 7x_{31} + 6x_{32} + 9x_{33}$$

(toda máquina recibe tarea) $$\sum_{j=1}^{3} x_{ij} = 1 \qquad \forall i = 1,\dots,3$$

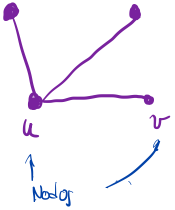
(toda tarea es asignada) $$\sum_{i=1}^{3} x_{ij} = 1 \qquad \forall j = 1,\dots,3$$

$$x_{ij} \in \{0,1\} \qquad \forall i,j$$

Modelos basados en grafos

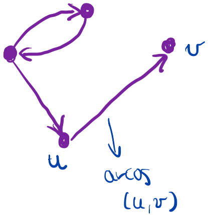
Definición

Un grafo simple, o no-dirigido, es un par $G = (V, E)$ donde cada elemento de E es un par $\{u, v\}$ con $u, v \in V$. V son los vértices o nodos y E son las aristas.



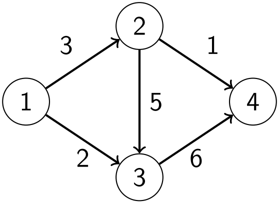
Definición

Un grafo dirigido es un par $G = (V, A)$ donde cada elemento de A es un par *ordenado* (u, v) con $u, v \in V$. V son los vértices o nodos, y A son los arcos.



Flujo máximo

Supongamos que tenemos el siguiente grafo dirigido:



En cada arco puede transitar un flujo (vehículos, datos, etc), con un máximo indicado en el arco correspondiente. Nos gustaría enviar la mayor cantidad de flujo posible desde 1 hasta 4.

Variables: $x_{u,v}$ = Cantidad de flujo en arco (u,v)

Es decir: $x_{12}, x_{13}, x_{23}, x_{24}, x_{34}$

Flujo máximo

$$\max \quad x_{12} + x_{13}$$ } Flujo saliente de 1

(conservación de flujo)
$$\left\{\begin{array}{l} x_{12} - x_{23} - x_{24} = 0 \\ x_{13} + x_{23} - x_{34} = 0 \end{array}\right.$$

(capacidades)
$$\left\{\begin{array}{l} x_{12} \leq 3 \\ x_{13} \leq 2 \\ x_{23} \leq 5 \\ x_{24} \leq 1 \\ x_{34} \leq 6 \end{array}\right.$$

$$x_{u,v} \geq 0$$