



upcomillas *es*

upcomillas *es*

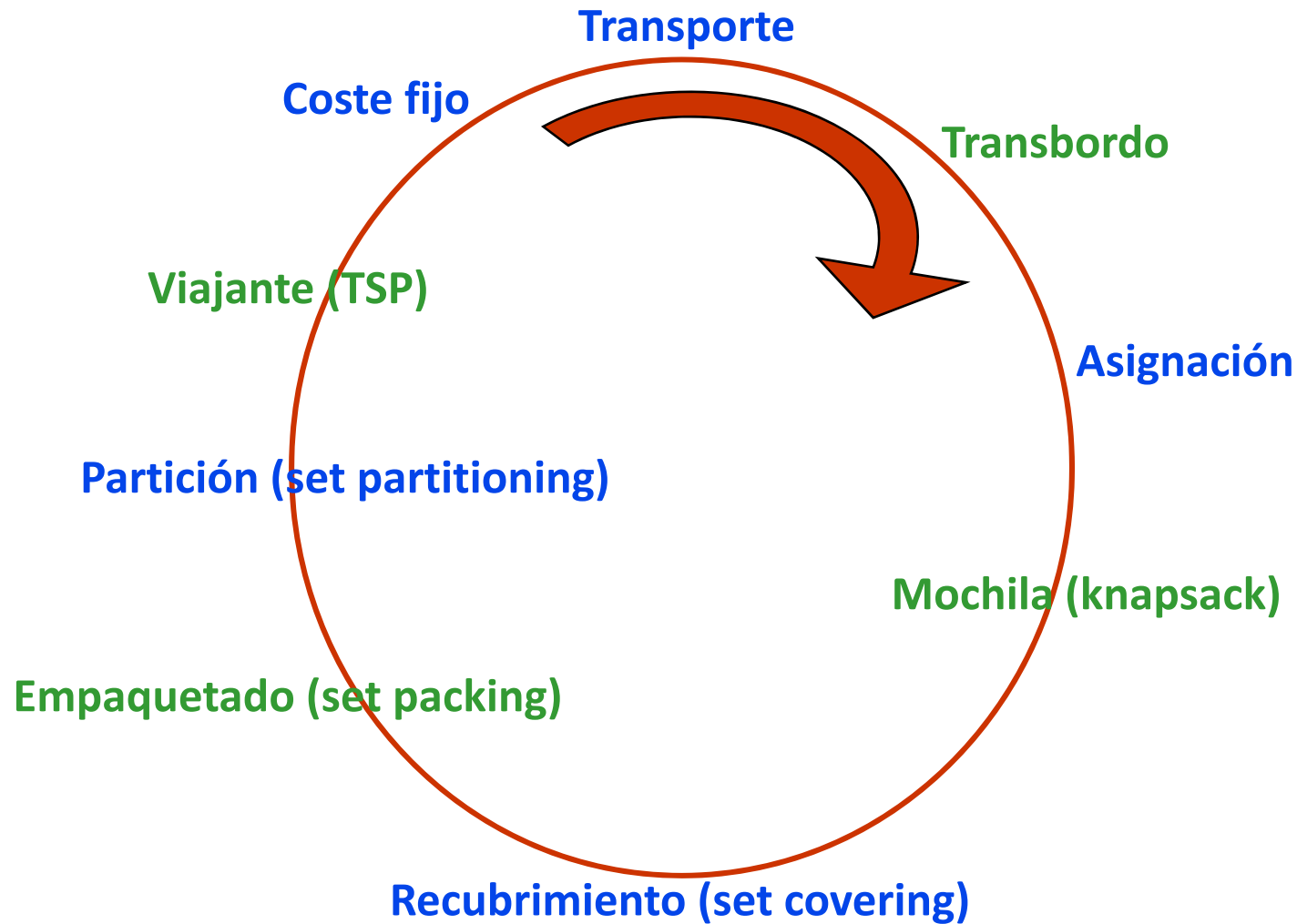
Gestión de operaciones

***Modelos de programación lineal
y entera***

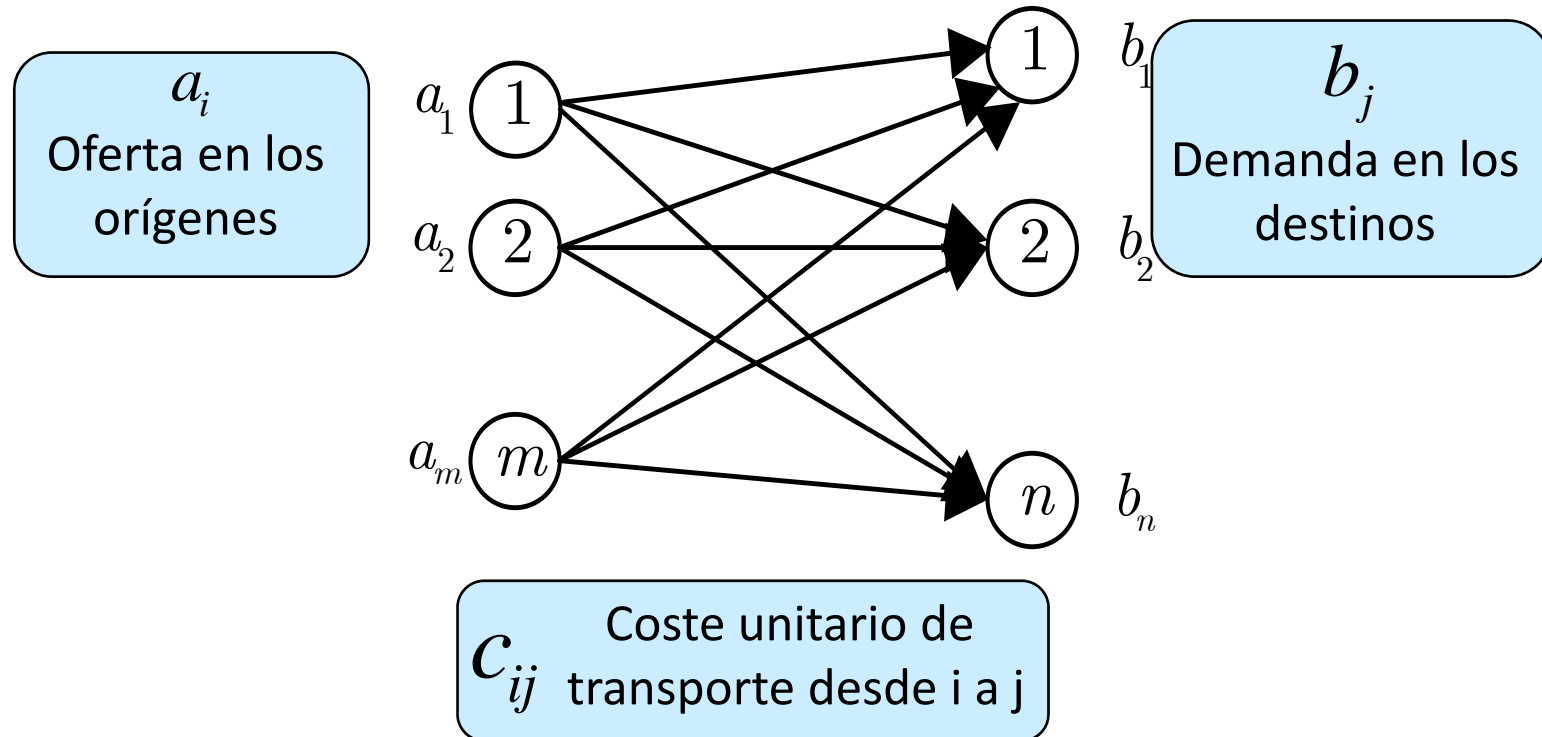
Pedro Sánchez

pedro.sanchez@upcomillas.es

Contenidos



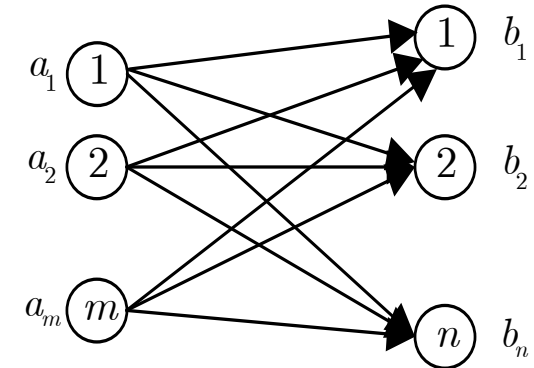
Problema del transporte (i)



Función objetivo: Minimizar el coste total de transporte desde unos orígenes a unos destinos

Problema del transporte (ii)

Variables: Cantidades de producto transportadas entre los orígenes y destinos



Todos los orígenes están conectados con todos los destinos

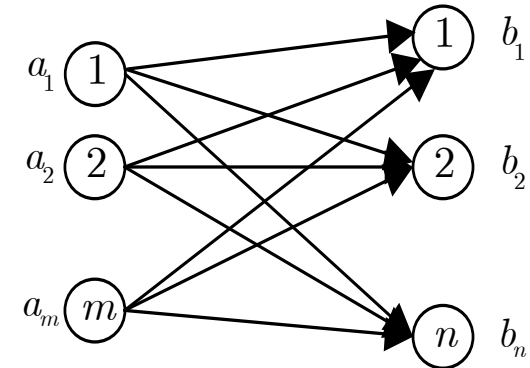
Restricciones:

- Cantidad disponible de producto en cada origen i (m restricciones)
- Demanda de cada destino j (n restricciones)

Problema del transporte (iii)

Formulación

$$\left\{ \begin{array}{l} \min_{x_{ij}} \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \sum_{j=1}^n x_{ij} = a_i \quad i = 1, \dots, m \\ \sum_{i=1}^m x_{ij} = b_j \quad j = 1, \dots, n \\ x_{ij} \geq 0 \end{array} \right.$$

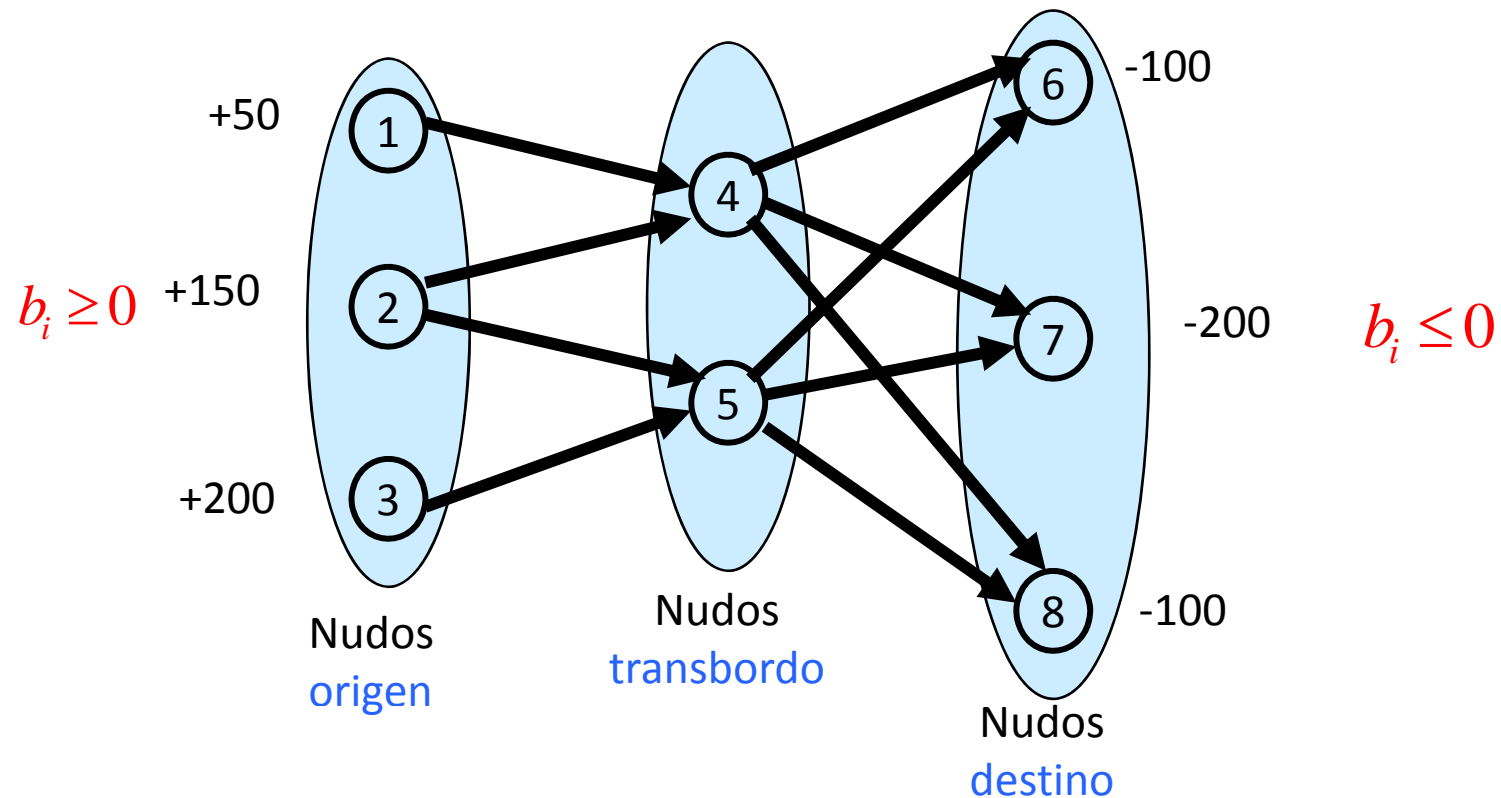


$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$$

- Esta formulación implica que la oferta es **igual** a la demanda
- Si $\sum_{i=1}^m a_i > \sum_{j=1}^n b_j$ se añade un **sumidero** universal con coste nulo
- Si $\sum_{i=1}^m a_i < \sum_{j=1}^n b_j$ se añade una **fuelle** universal con coste elevado
- Por su estructura de coeficientes se resuelve como un problema **lineal**

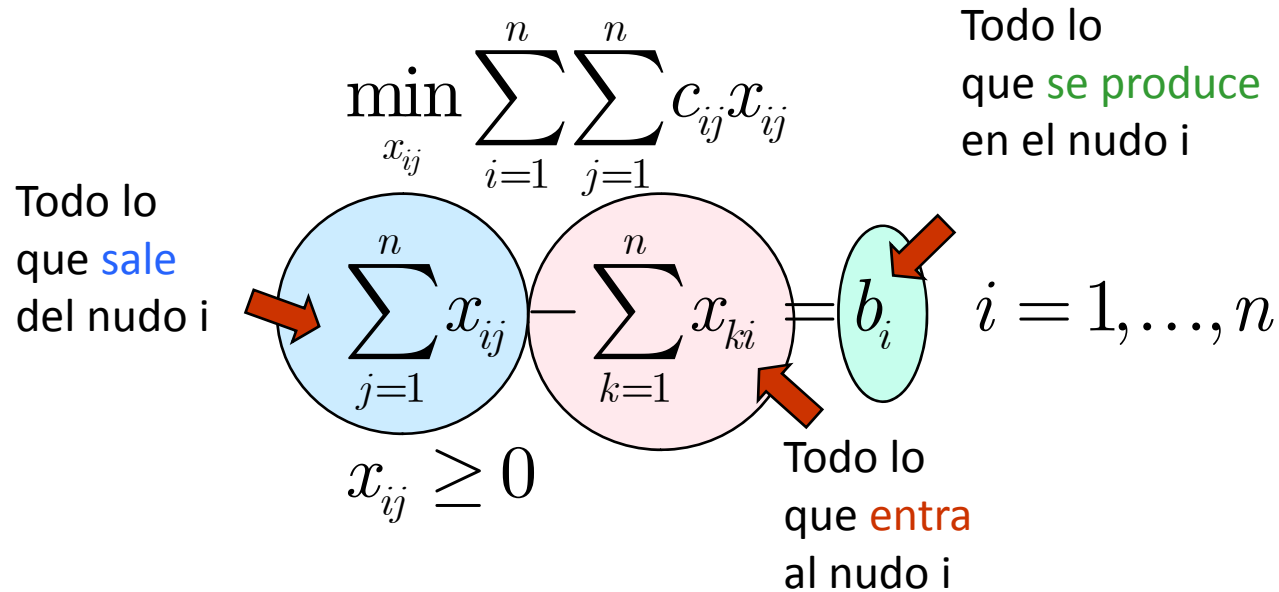
Problema del transbordo (i)

Consiste en determinar las cantidades óptimas a transportar desde sus orígenes a destinos pasando por puntos de transbordo intermedios



Problema del transbordo (ii)

Se minimizan los costes de transporte



Problema del transbordo (iii)

Extensiones del modelo con excesos y defectos de producto

$$\min_{x_{ij}} \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} + \sum_{i=1}^n p_i \cdot (e_i + d_i)$$

$$\sum_{j=1}^n x_{ij} - \sum_{k=1}^n x_{ki} = b_i - e_i + d_i \quad i = 1, \dots, n$$

$$x_{ij} \geq 0$$

p_i : penalización por exceso y por defecto

e_i : variable por exceso de producto en origen i

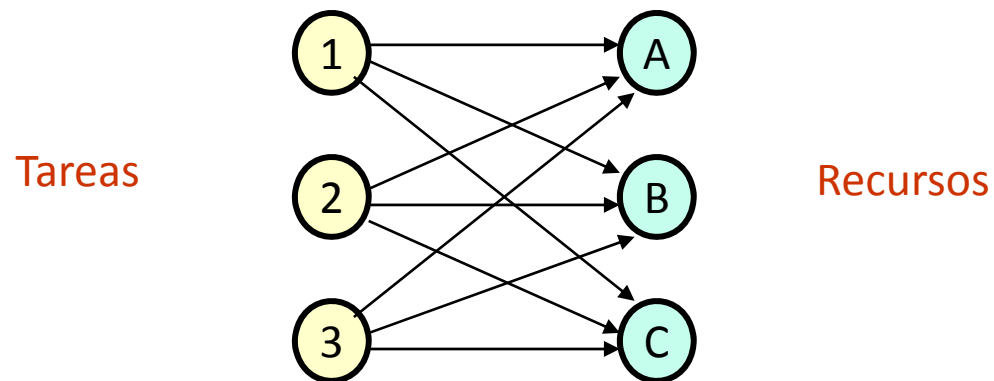
d_i : variable por defecto de producto en destino i

Problema de asignación (i)

- Asignar n tareas a n recursos (personas, máquinas,...)
- Es un caso particular del problema de transporte
- Es un problema con variables binarias de asignación
- Se resuelve como un problema de programación lineal

$$x_{ij} \begin{cases} 1 & \text{si se asigna la tarea } i \text{ al recurso } j \\ 0 & \text{si no se asigna la tarea } i \text{ al recurso } j \end{cases}$$

- Se denomina C_{ij} al coste de asignar la tarea i al recurso j



Problema de asignación (ii)

- Formulación matemática:

$$\min_{x_{ij}} \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$



Se **minimiza** el coste de asignación

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, \dots, n$$



Cada **tarea** ha de ser realizada

$$\sum_{i=1}^n x_{ij} = 1 \quad j = 1, \dots, n$$



Cada **recurso** es utilizado una sola vez

$$x_{ij} \geq 0$$

Problema de la mochila

- Se trata de maximizar el valor total de la elección de un conjunto de n proyectos sin sobrepasar el presupuesto b disponible
- V_j : valor de cada proyecto j
- C_j : coste de cada proyecto j

Variables de decisión: $x_j \begin{cases} 1 & \text{si se elige el proyecto } j \\ 0 & \text{en cualquier otro caso} \end{cases}$

Formulación matemática

$$\left\{ \begin{array}{l} \max_{x_j} \sum_{j=1}^n v_j x_j \quad \longrightarrow \quad \text{Maximizar el valor total} \\ \sum_{j=1}^n c_j x_j \leq b \quad \longrightarrow \quad \text{No sobrepasar ppto} \\ x_j \in \{0,1\} \quad \longrightarrow \quad \text{Variables binarias} \end{array} \right.$$

Problema del recubrimiento (set covering) (i)

- Existen m características y n combinaciones de características
- Una combinación elegida implica realizar todas las características de dicha combinación
- **Datos:**

c_j : coste de elegir la combinación j

a_{ij} : elementos de la matriz de pertenencia

$$a_{ij} \begin{cases} 1 & \text{si } i \text{ pertenece a } j \\ 0 & \text{si } i \text{ no pertenece a } j \end{cases}$$

- **Variables:**

$$x_j \begin{cases} 1 & \text{si se elige la combinación } j \\ 0 & \text{en cualquier otro caso} \end{cases}$$

Problema del recubrimiento (set covering) (ii)

Formulación matemática:

$$\min_{x_j} \sum_{j=1}^n c_j x_j \longrightarrow \text{Minimizar el coste de asignación}$$

$$\sum_{j=1}^n a_{ij} x_j \geq 1 \quad i = 1, \dots, m \longrightarrow \text{Todas las características seleccionadas}$$

$$x_j \in \{0, 1\} \longrightarrow \text{Variables binarias}$$

Problema del recubrimiento (set covering) (iii)

Ejemplo:

Una compañía aérea quiere cubrir todos sus vuelos asignando tres tripulaciones con base en San Francisco a las 12 secuencias de vuelos que se indican en la siguiente tabla

	Secuencias factibles											
	1	2	3	4	5	6	7	8	9	10	11	12
SF – LA	1			1			1			1		
SF – Denver		1			1			1			1	
SF – Seattle			1			1			1			1
LA – Chicago				2			2		3	2		3
LA – SF	2					3				5	5	
Chicago – Denver				3	3				4			
Chicago – Seattle							3	3		3	3	4
Denver – SF		2		4	4				5			
Denver – Chicago					2			2			2	
Seattle – SF			2				4	4				5
Seattle – LA						2			2	4	4	2
Coste (Mpta)	2	3	4	6	7	5	7	8	9	9	8	9

← Coste

Esta asignación ha de realizarse minimizando el coste

Problema del recubrimiento (set covering) (iv)

Ejemplo (Cont.):

$$\min 2x_1 + 3x_2 + 4x_3 + 6x_4 + 7x_5 + 5x_6 + 7x_7 + 8x_8 + 9x_9 + 9x_{10} + 8x_{11} + 9x_{12}$$

$$x_1 + x_4 + x_7 + x_{10} \geq 1$$

$$x_2 + x_5 + x_8 + x_{11} \geq 1$$

$$x_3 + x_6 + x_9 + x_{12} \geq 1$$

⋮

$$\sum_{j=1}^{12} x_j = 3 \quad x_j \in \{0,1\} \forall j$$

Soluciones óptimas:
$$\begin{cases} x_3 = x_4 = x_{11} = 1 \\ x_1 = x_5 = x_{12} = 1 \end{cases} \quad \text{Coste: 18 k€}$$

Problema del empaquetado (i)

- Se pueden realizar m proyectos divididos en n paquetes
- Maximizar el beneficio total de manera que cada proyecto i no sea elegido más de una vez
- Datos:

C_j : beneficio de elegir el paquete j

a_{ij} : matriz de pertenencia del proyecto i a cada paquete j

- Variables:

$$x_j \begin{cases} 1 & \text{si se elige el paquete } j \\ 0 & \text{en cualquier otro caso} \end{cases}$$

Problema del empaquetado (ii)

Formulación matemática:

$$\max_{x_j} \sum_{j=1}^n c_j x_j \quad \longrightarrow \quad \text{Maximizar el beneficio total}$$

$$\sum_{j=1}^n a_{ij} x_j \leq 1 \quad i = 1, \dots, m \quad \longrightarrow \quad \text{No hacer dos veces un mismo proyecto}$$

$$x_j \in \{0, 1\} \quad \longrightarrow \quad \text{Variables binarias}$$

Problema de partición

Es igual que el problema del empaquetado salvo que todas las características o proyectos deben ser elegidas

$$\max_{x_j} \sum_{j=1}^n c_j x_j \quad \longrightarrow \quad \text{Maximizar el beneficio total}$$

$$\sum_{j=1}^n a_{ij} x_j = 1 \quad i = 1, \dots, m \quad \longrightarrow \quad \text{Hacer todos los proyectos}$$

$$x_j \in \{0, 1\} \quad \longrightarrow \quad \text{Variables binarias}$$

Problema del viajante (TSP) (i)

Consiste en hacer un recorrido que pase por n ciudades sin repetir ninguna y volver a la ciudad de partida, haciendo que la distancia (coste) sea mínima

FORMULACIÓN 1

$$\min_{x_{ij}} \sum_{i,j} c_{ij} x_{ij}$$



Minimizar la distancia total

$$\sum_i x_{ij} = 1 \quad \forall j$$



Se llega a todas las ciudades

$$\sum_j x_{ij} = 1 \quad \forall i$$



Se sale de todas las ciudades

$$\sum_{i,j \in U} x_{ij} \leq \text{Card}(U) - 1 \quad \forall U \subset \{1, \dots, n\} \quad / \quad 2 \leq \text{Card}(U) \leq n - 2$$



Evitar recorridos de orden inferior

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



Variables binarias de trayecto entre i y j

Problema del viajante (TSP) (ii)

FORMULACIÓN 2

$$x_{ijk} = \begin{cases} 1 & \text{si se va de la ciudad } i \text{ a la ciudad } j \text{ en el tramo } k \text{ de recorrido} \\ 0 & \text{en otro caso} \end{cases}$$

$$\min \sum_{i,j,k} c_{ij} x_{ijk} \quad \longrightarrow \quad \text{Minimizar la distancia total}$$

$$\sum_{j,k} x_{ijk} = 1 \quad \forall i \quad \longrightarrow \quad \text{Se sale de todas las ciudades}$$

$$\sum_{i,k} x_{ijk} = 1 \quad \forall j \quad \longrightarrow \quad \text{Se llega a todas las ciudades}$$

$$\sum_{i,j} x_{ijk} = 1 \quad \forall k \quad \longrightarrow \quad \text{Se completan todos los tramos}$$

$$\sum_i x_{ijk} = \sum_r x_{jrk+1} \quad \forall j, k \quad \longrightarrow \quad \text{Realización tramos consecutivos}$$

$$x_{ijk} \in \{0, 1\} \quad \longrightarrow \quad \text{Variables binarias}$$

Problema de coste fijo

Una variable x_j tiene un coste con un término fijo k_j si dicha variable toma valor estrictamente positivo y un término variable c_j

$$f_j(x_j) = \begin{cases} 0 & x_j = 0 \\ k_j + c_j x_j & x_j > 0 \end{cases}$$

Este tipo de coste se modela con una variable binaria auxiliar

$$y_j \in \{0,1\} \quad x_j \leq M \cdot y_j \quad y_j = \begin{cases} 1 & x_j > 0 \\ 0 & x_j = 0 \end{cases}$$

Función objetivo: $\min_{x_j, y_j} \sum_{j=1}^n f_j(x_j) = \sum_{j=1}^n (k_j y_j + c_j x_j)$