UNIVERSIDAD PONTIFICIA
ICAI ICADE
COMILLAS
MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

Modelado en optimización

José María Ferrer Caja
Universidad Pontificia Comillas

Optimización

- Encontrar el valor que deben tomar las variables del problema para hacer óptima la función objetivo, y de forma que se satisfagan las restricciones

Componentes de un problema de optimización

- Función objetivo
  - ✓ Medida cuantitativa del funcionamiento del sistema que se desea optimizar (maximizar o minimizar)
- Variables
  - ✓ Representan las decisiones que se pueden adoptar y de las que depende el valor de la función objetivo
  - ✓ Variables de control y de estado
- Restricciones
  - ✓ Relaciones que las variables deben satisfacer
  - ✓ Igualdades y desigualdades

Clasificación de modelos

- Función objetivo
  - ✓ Lineal, cuadrática, no lineal, no diferenciable, multiobjetivo, estocástica, sin función objetivo
- Restricciones
  - ✓ Sin restricciones, lineales, no lineales, no diferenciables, acotadas, disyuntivas, probabilísticas
- Variables
  - ✓ Continuas, enteras, binarias

Modelo

- Representación simplificada de la realidad, generalmente de forma matemática, que facilita su comprensión y el estudio de su comportamiento
- Un buen modelo debe mantener un equilibrio entre sencillez y capacidad de representación
- Se requiere una interacción constante entre el modelador (creador del modelo) y el experto (conocedor del problema real)
- Es a la vez una ciencia y un arte

Etapas en el desarrollo de un modelo

- Identificación del problema
- Especificación matemática y formulación
- Resolución
- Verificación, validación y refinamiento
- Interpretación y análisis de resultados
- Implantación, documentación y mantenimiento

- El tiempo empleado en realizar correctamente una etapa facilitará de forma notable las etapas sucesivas

Identificación del problema

- Recolección de información relevante
- Planteamiento general del problema
  - ✓ Qué se quiere optimizar
  - ✓ De qué alternativas se dispone
  - ✓ Qué limitaciones se tienen
- Interpretación y traducción a términos precisos de los datos y elementos del problema
- Fase fundamental para que las futuras decisiones sean útiles

Especificación matemática y formulación

- Definición en términos matemáticos adecuados de los elementos del problema
  - ✓ Función objetivo
  - ✓ Variables
  - ✓ Restricciones
  - ✓ Parámetros
- Identificación del tipo(s) de modelo general que se puede aplicar
- Formulación clara y elegante
- Análisis del tamaño y estructura del problema formulado

Resolución

- Elección del tipo de método de solución y de algoritmos adecuados
- Implementación (en un lenguaje informático adecuado) de los algoritmos elegidos
- Obtención de la solución óptima o de soluciones suficientemente satisfactorias
- El tiempo de resolución dependerá de la formulación propuesta

Verificación, validación y refinamiento

- Comprobación de coherencia con la realidad
- Detección y corrección de errores de codificación
- Mejora y ampliación por nuevas necesidades
  - ✓ En la definición
  - ✓ En la formulación
  - ✓ En la implementación

Interpretación y análisis de resultados

- Análisis de sensibilidad de la solución obtenida, frente a cambios en los parámetros
- Detección de soluciones robustas

Implantación, documentación y mantenimiento

- Documentación clara, precisa y completa
- El código debe estar escrito de forma ordenada y debe incluir comentarios que faciliten las operaciones de mantenimiento
- Elaboración de un manual de usuario con especificaciones matemáticas e informáticas
- Formación de usuarios

Ejemplo de la dieta: Planteamiento

- La alimentación diaria de una ternera debe contener al menos:
  - ✓700 g de proteínas
  - ✓28 g de calcio
  - ✓150 mg de vitaminas
- Se dispone de pienso y forraje con coste por kg de 30 y 35 céntimos de euro
- La composición nutritiva por kg de alimento es:

	Proteínas (g)	Calcio (g)	Vitaminas (mg)
Pienso	30	2	10
Forraje	45	1	5

- Se trata de determinar la cantidad diaria de cada alimento para minimizar el coste total de la alimentación

Ejemplo de la dieta: Formulación genérica

- Índices:
  - $i \rightarrow$ tipo de alimento (pienso y forraje)
  - $j \rightarrow$ tipo de nutriente (proteínas, calcio y vitaminas)
- Parámetros:
  - $b_j \rightarrow$ cantidad mínima diaria requerida del nutriente j
  - $a_{ij} \rightarrow$ cantidad de nutriente j por kg de alimento i
  - $c_i \rightarrow$ coste por kg del alimento i
- Variables:
  - $x_i \rightarrow$ cantidad diaria de alimento i por ternera
- Función objetivo: $\min_{x_i} \sum_i c_i x_i$
- Restricciones:
  - De satisfacción de la cantidad de nutrientes $\sum_i a_{ij} x_i \geq b_j \quad \forall j$
  - De no negatividad $x_i \geq 0 \ \forall i$

Ejemplo de la dieta: Formulación numérica

$$\min_{x_1, x_2} 30x_1 + 35x_2$$

$$30x_1 + 45x_2 \geq 700 \qquad \text{(proteínas)}$$

$$2x_1 + x_2 \geq 28 \qquad \text{(calcio)}$$

$$10x_1 + 5x_2 \geq 150 \qquad \text{(vitaminas)}$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

Modelos generales de optimización: Programación lineal (LP)

$$\min_x c^T x$$
$$Ax = b$$
$$x \geq 0$$
$$x \in \mathbb{R}^n, c \in \mathbb{R}^n, A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$$

$\Leftrightarrow$

$$\begin{aligned}
&\min \ z = c_1x_1 + c_2x_2 + \ldots + c_nx_n \\
&s.a \\
&a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\
&\vdots \\
&a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m \\
&x_1, x_2, \ldots, x_n \geq 0
\end{aligned}$$

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$ vector de variables

$$c = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}$$ vector de coeficientes de coste

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$ matriz de restricciones

$$b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$$ vector de demandas

Modelos generales de optimización: Programación entera (IP)

$$\boxed{\begin{array}{l}\min\limits_{x} c^T x + d^T y \\ Ax + By = b \\ x, y \geq 0 \\ x \in \mathbb{Z}^n, y \in \mathbb{R}^l, c \in \mathbb{R}^n, d \in \mathbb{R}^l \\ A \in \mathbb{R}^{m\times n}, B \in \mathbb{R}^{m\times l}, b \in \mathbb{R}^m\end{array}}$$

✓Si $l = 0$ → Programación entera pura (PIP)
✓Si $l > 0$ → Programación entera mixta (MIP)

Modelos generales de optimización: Programación binaria (BIP)

$$\min_x c^T x$$
$$Ax = b$$
$$x \geq 0$$
$$x \in \{0,1\}^n, c \in \mathbb{R}^n, A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$$

Modelos generales de optimización: Programación cuadrática (QP)

$$
\begin{aligned}
&\min_x c^T x + \frac{1}{2} x^T Q x \\
&Ax = b \\
&x \geq 0 \\
&x \in \mathbb{R}^n, c \in \mathbb{R}^n, A \in \mathbb{R}^{m\times n} \\
&Q \in \mathbb{R}^{n\times n}, b \in \mathbb{R}^m
\end{aligned}
$$

Modelos generales de optimización: Programación no lineal (NLP)

$$\boxed{\begin{array}{l} \min\limits_{x} f(x) \\ g(x) = 0 \\ h(x) \leq 0 \\ l \leq x \leq u \\ f : \mathbb{R}^n \longrightarrow \mathbb{R} \\ g : \mathbb{R}^n \longrightarrow \mathbb{R}^m \\ h : \mathbb{R}^n \longrightarrow \mathbb{R}^k \end{array}}$$

Modelos generales de optimización: Programación multiobjetivo

$$
\begin{aligned}
&\min_x (f_1(x), \ldots, f_k(x)) \\
&Ax = b \\
&x \geq 0 \\
&x \in \mathbb{R}^n, c \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m \\
&f_i(x) : \mathbb{R}^n \to \mathbb{R}
\end{aligned}
$$

Modelos generales de optimización: Especiales

Optimización no lineal sin restricciones	$\min_x f(x)$ $f : \mathbb{R}^n \longrightarrow \mathbb{R}$
Ajuste no lineal mínimo cuadrático	
Problema mixto complementario (MCP)	$xF(x) = 0$ $x \in \mathbb{R}^n$ $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$

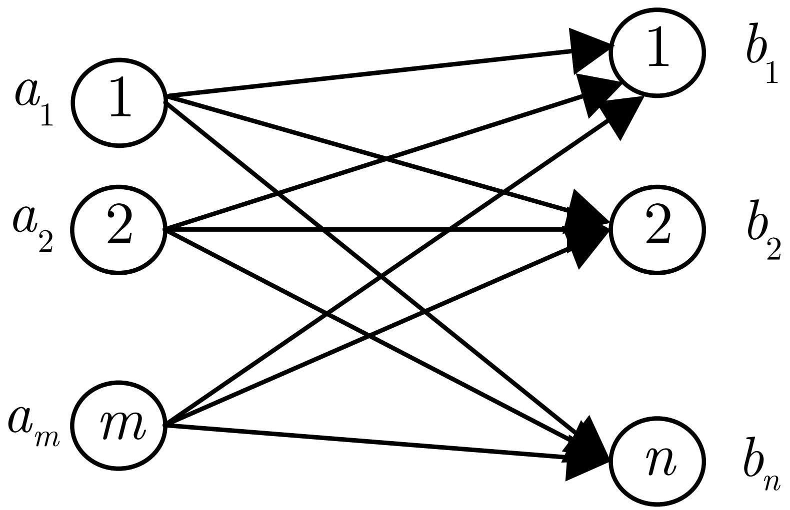
Modelos específicos de programación lineal y entera

- Problema del transporte
- Problema del transbordo
- Problema de asignación
- Problema de la mochila (knapsack)
- Problema del recubrimiento (set-covering)
- Problema del empaquetado (set-packing)
- Problema de la partición (set-partitioning)
- Problema del viajante (TSP)

Problema del transporte: Definición

- Minimizar el coste total de transporte de un producto desde los orígenes a los destinos, satisfaciendo la demanda de cada destino sin superar la oferta disponible en cada origen

a_i
Oferta en el origen i



b_j
Demanda en el destino j

c_{ij}
Coste unitario de transporte del origen i al destino j

Se supone que la oferta es igual a la demanda: $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$

Problema del transporte: Formulación

- Variables de decisión

  x_{ij} = unidades transportadas desde el origen *i* al destino *j*

$$
\begin{aligned}
&\min_{x_{ij}} \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \\
&s.a \\
&\sum_{i=1}^{m} x_{ij} = b_j \qquad j = 1,\ldots,n \\
&\sum_{j=1}^{n} x_{ij} = a_i \qquad i = 1,\ldots,m \\
&x_{ij} \geq 0 \qquad i = 1,\ldots,m, j = 1,\ldots,n
\end{aligned}
$$

Problema del transporte: Propiedades

- ✓ 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 fuente universal con coste elevado

- ✓ La matriz de restricciones es totalmente unimodular: todos los determinantes de las submatrices valen 0, 1 ó -1
- ✓ Por lo tanto, si los coeficientes de oferta a_i y demanda b_j son enteros la solución óptima es un vector entero $\rightarrow \quad x_{ij}^* \in \mathbb{Z} \ \forall i, j$

Problema del transbordo: Definición

- Transportar un producto en una red satisfaciendo la demanda, con el mínimo coste posible

$b_i \rightarrow$ Cantidad de producto disponible en el nodo i
- ✓ oferta: $b_i > 0$
- ✓ demanda: $b_i < 0$
- ✓ transbordo: $b_i = 0$

$c_{ij} \rightarrow$ Coste unitario de transporte del nodo i al nodo j

Se supone que la oferta es igual a la demanda: $\sum_{i=1}^{m} b_i = 0$

Problema del transbordo: Formulación

- Variables de decisión

  x_{ij} = unidades transportadas desde el nodo *i* al nodo *j*

conservación del flujo

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

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ij} - \sum_{\substack{k=1 \\ k \neq i}}^{n} x_{ki} = b_i \qquad i = 1, \ldots, n$$

$$x_{ij} \geq 0 \qquad i, j = 1, \ldots, n, i \neq j$$

✓ Matriz totalmente unimodular: $b_i \in \mathbb{Z} \ \forall i \Rightarrow x^*_{ij} \in \mathbb{Z} \ \forall i, j$

✓ El problema del transbordo generaliza al problema del transporte

Problema de asignación

- Asignar *n* tareas a *n* máquinas de forma que el coste total sea mínimo

  c_{ij} → Coste de asignar la tarea *i* a la máquina *j*

- Variables $x_{ij} = \begin{cases} 1 & \text{si se asigna la tarea } i \text{ a la máquina } j \\ 0 & \text{si no} \end{cases}$

- Formulación

$$\boxed{\begin{aligned} &\min_{x_{ij}} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \\ &\sum_{j=1}^{n} x_{ij} = 1 \quad i = 1, \ldots, n \\ &\sum_{i=1}^{n} x_{ij} = 1 \quad j = 1, \ldots, n \\ &x_{ij} \in \{0,1\} \quad \forall i, j \end{aligned}}$$

Cada tarea ha de ser realizada por una máquina

Cada máquina ha de realizar una tarea

✓ Caso particular del problema del transporte

✓ La restricción $x_{ij} \in \{0,1\}$ puede sustituirse por $x_{ij} \geq 0$

Problema de la mochila

- Elegir objetos de los n disponibles de forma que el valor total sea máximo, sin sobrepasar el volumen disponible

  $c_j \rightarrow$ Volumen que ocupa el objeto j

  $v_j \rightarrow$ Valor del objeto j

  $b \rightarrow$ Volumen total disponible

- Variables

$$x_j = \begin{cases} 1 & \text{si se elige el objeto } j \\ 0 & \text{si no} \end{cases}$$

- Formulación

$$\boxed{\begin{aligned} &\max_{x_j} \sum_{j=1}^{n} v_j x_j \\ &\sum_{j=1}^{n} c_j x_j \le b \\ &x_j \in \{0,1\} \quad \forall j \end{aligned}}$$

Problema del recubrimiento

- Se dispone de un conjunto de *m* elementos y una colección de *n* subconjuntos
- Se desea elegir subconjuntos que cubran todos los elementos al menos una vez, y con el mínimo coste

  $c_j \rightarrow$ Coste del subconjunto *j*

  $a_{ij} \rightarrow$ Indica si el elemento *i* pertenece al subconjunto *j* (1 sí, 0 no)

- Variables $x_j = \begin{cases} 1 & \text{si se elige el subconjunto } j \\ 0 & \text{si no} \end{cases}$

- Formulación

Cada elemento seleccionado al menos una vez

$$\boxed{\begin{aligned} &\min_{x_j} \sum_{j=1}^{n} c_j x_j \\ &\sum_{j=1}^{n} a_{ij} x_j \geq 1 \quad i = 1, \ldots, m \\ &x_j \in \{0,1\} \quad j = 1, \ldots, n \end{aligned}}$$

Problema del recubrimiento: Ejemplo

- Una compañía aérea con base en San Francisco quiere cubrir todos sus vuelos asignando tres tripulaciones a algunas de las 12 secuencias factibles de vuelos que se indican en la siguiente tabla.
- El objetivo es minimizar el coste de la asignación. Los costes de las diferentes secuencias aparecen en la última fila:

	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 (M€)	2	3	4	6	7	5	7	8	9	9	8	9

Problema del recubrimiento: Ejemplo

- Variables: $x_j = \begin{cases} 1 & \text{si se asigna la secuencia } j \\ 0 & \text{en cualquier otro caso} \end{cases}$

- Formulación:

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

$$x_1 + x_4 + x_7 + x_{10} \geq 1$$ (SF-LA)

$$x_2 + x_5 + x_8 + x_{11} \geq 1$$ (SF-Denver)

$$x_3 + x_6 + x_9 + x_{12} \geq 1$$ (SF-Seattle)

$$\vdots$$

$$\sum_{j=1}^{12} x_j = 3$$ (Dispone de tres tripulaciones)

$$x_j \in \{0,1\} \quad j = 1,\ldots,12$$

$$x_3 = x_4 = x_{11} = 1$$

$$x_1 = x_5 = x_{12} = 1$$

Problema del empaquetado

- Se dispone de un conjunto de *m* elementos y una colección de *n* subconjuntos
- Se desea elegir los subconjuntos que den el máximo beneficio total, sin que ningún elemento aparezca más de una vez

  $c_j \rightarrow$ Beneficio del subconjunto *j*

  $a_{ij} \rightarrow$ Indica si el elemento *i* pertenece al subconjunto *j* (1 sí, 0 no)

- Variables $$x_j = \begin{cases} 1 & \text{si se elige el subconjunto } j \\ 0 & \text{si no} \end{cases}$$

- Formulación

Cada elemento seleccionado como mucho una vez

$$\boxed{\begin{aligned} &\max_{x_j} \sum_{j=1}^{n} c_j x_j \\ &\sum_{j=1}^{n} a_{ij} x_j \le 1 \quad i = 1,\dots,m \\ &x_j \in \{0,1\} \quad j = 1,\dots,n \end{aligned}}$$

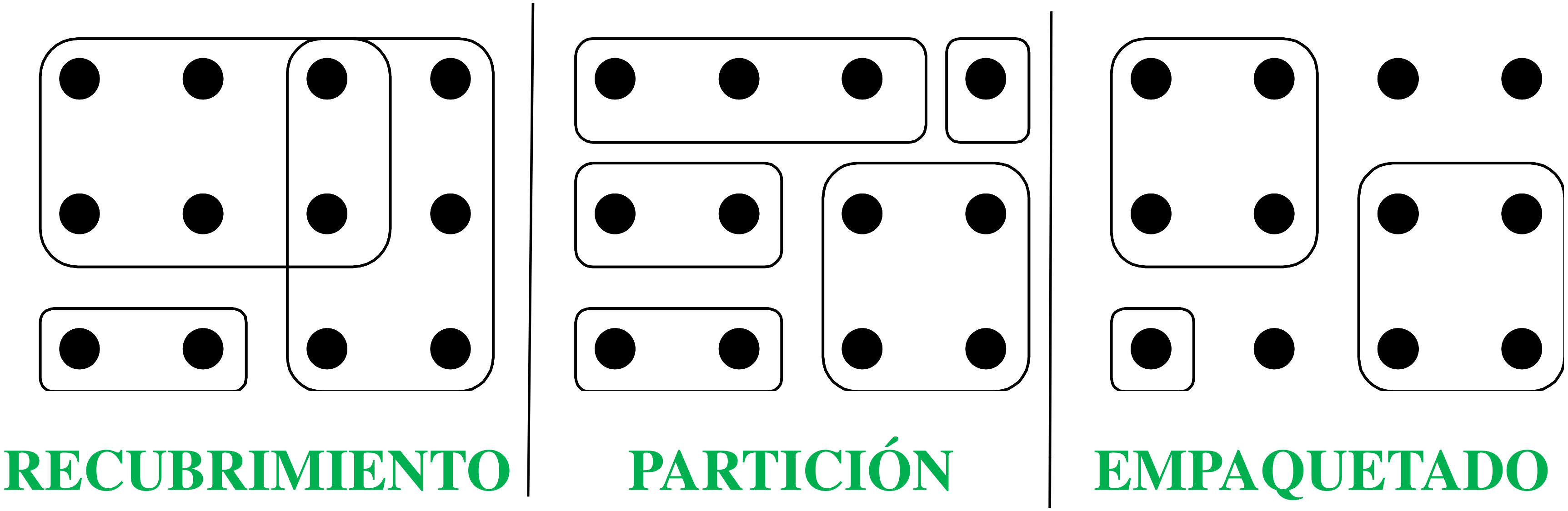
Problema de la partición

- Análogo al problema del empaquetado, pero seleccionando exactamente una vez cada elemento

Cada elemento seleccionado una vez

$$
\begin{aligned}
&\max_{x_j} \sum_{j=1}^{n} c_j x_j \\
&\sum_{j=1}^{n} a_{ij} x_j = 1 \quad i = 1,\ldots,m \\
&x_j \in \{0,1\} \quad j = 1,\ldots,n
\end{aligned}
$$

Recubrimiento, partición, empaquetado



Problema del viajante: Formulación 1

- Realizar un circuito que pase por n ciudades sin repetir ninguna (volviendo a la ciudad de partida) de manera que la distancia (o tiempo o coste) total sea mínima

  $c_{ij} \rightarrow$ Distancia de la ciudad i a la ciudad j

- Variables: $$x_{ij} = \begin{cases} 1 & \text{si se va de la ciudad } i \text{ a la ciudad } j \\ 0 & \text{en otro caso} \end{cases}$$

- Formulación:

A cada ciudad se llega una vez

De cada ciudad se sale una vez

No se permiten subciclos

$$\min_{x_{ij}} \sum_{i,j} c_{ij} x_{ij}$$
$$\sum_{i} x_{ij} = 1 \quad \forall j = 1,\ldots,n$$
$$\sum_{j} x_{ij} = 1 \quad \forall i = 1,\ldots,n$$
$$\sum_{i,j \in U} x_{ij} \le \operatorname{card}(U) - 1 \quad \forall U \subset \{1,\ldots,n\}, 2 \le \operatorname{card}(U) \le n-2$$
$$x_{ij} \in \{0,1\} \quad \forall i,j = 1,\ldots,n$$

Problema del viajante: Formulación 2

- **Variables**: $x_{ijk} = \begin{cases} 1 & \text{si se va de la ciudad } i \text{ a la ciudad } j \text{ en la etapa } k \text{ del circuito} \\ 0 & \text{en otro caso} \end{cases}$

- **Formulación**:

De cada ciudad se sale una vez

A cada ciudad se llega una vez

En cada etapa se recorre un tramo

En cada etapa se sale de la ciudad a la que se ha llegado en la etapa anterior

$$\boxed{\begin{aligned} &\min_{x_{ijk}} \sum_{i,j,k} c_{ij} x_{ijk} \\ &\sum_{j,k} x_{ijk} = 1 \quad \forall i = 1,\dots,n \\ &\sum_{i,k} x_{ijk} = 1 \quad \forall j = 1,\dots,n \\ &\sum_{i,j} x_{ijk} = 1 \quad \forall k = 1,\dots,n \\ &\sum_{i} x_{ijk} = \sum_{r} x_{jrk+1} \quad \forall j,k = 1,\dots,n \\ &x_{ijk} \in \{0,1\} \quad \forall i,j,k = 1,\dots,n \end{aligned}}$$

Modelado con variables binarias

- Conversión de entera a binarias
- Disyunciones
- Disyunciones por bloques
- Cumplimiento de un número de ecuaciones
- Selección entre varios valores
- Implicaciones entre variables binarias y restricciones
- Implicaciones
- Relaciones sencillas entre variables binarias
- Productos

Conversión de entera a binarias

- Se quiere descomponer la variable entera x en variables binarias y_1, y_2 , …
  - ✓ Se busca una cota superior u

$$0 \le x \le u$$

  - ✓ Se determina N tal que

$$2^N \le u \le 2^{N+1}$$

  - ✓ La descomposición es

$$x = \sum_{i=0}^{N} 2^i y_i$$

  - ✓ Se ha de sustituir en el modelo x por dicha expresión, añadiendo

$$y_i \in \{0,1\} \ \forall i$$

Tabla de equivalencias lógicas

$P \rightarrow Q$	no P ó Q
$P \leftrightarrow Q$	$(P \rightarrow Q)$ y $(Q \rightarrow P)$
$P \rightarrow (Q$ y $R)$	$(P \rightarrow Q)$ y $(P \rightarrow R)$
$P \rightarrow (Q$ ó $R)$	$(P \rightarrow Q)$ ó $(P \rightarrow R)$
$(P$ y $Q) \rightarrow R$	$(P \rightarrow R)$ ó $(Q \rightarrow R)$
$(P$ ó $Q) \rightarrow R$	$(P \rightarrow R)$ y $(Q \rightarrow R)$
no (P ó Q)	no P y no Q
no (P y Q)	no P ó no Q

Disyunciones

- $f(x) \le 0$ ó $g(x) \le 0$ se modela mediante

$$\begin{cases} f(x) \le M\delta \\ g(x) \le M(1-\delta) \end{cases} \quad \delta \in \{0,1\}$$

  - ✓ *M* es suficientemente grande. Lo ideal es que la *M* de cada restricción sea la cota más ajustada para cada restricción
  - ✓ La implicación $f(x) > 0 \Rightarrow g(x) \le 0$ es equivalente a $f(x) \le 0$ ó $g(x) \le 0$

- Ejemplo:

$$3x + 2y - z < 3 \Rightarrow x - 4y + z \ge 1$$

$$\Leftrightarrow$$

$$-3x - 2y + z + 3 > 0 \Rightarrow -x + 4y - z + 1 \le 0$$

$$\Leftrightarrow$$

$$-3x - 2y + z + 3 \le 0 \quad \text{ó} \quad -x + 4y - z + 1 \le 0$$

$$\Leftrightarrow$$

$$\begin{cases} -3x - 2y + z + 3 \le M\delta \\ -x + 4y - z + 1 \le M(1-\delta) \end{cases} \quad \delta \in \{0,1\}$$

Disyunciones por bloques

- $\begin{cases} f_1(x) \le 0 \\ f_2(x) \le 0 \\ \dots \\ f_k(x) \le 0 \end{cases}$ ó $\begin{cases} g_1(x) \le 0 \\ g_2(x) \le 0 \\ \dots \\ g_l(x) \le 0 \end{cases}$ se modela mediante

$$\begin{cases} f_1(x) \le M\delta \\ \dots \\ f_k(x) \le M\delta \\ g_1(x) \le M(1-\delta) \\ \dots \\ g_l(x) \le M(1-\delta) \end{cases} \qquad \delta \in \{0,1\}$$

✓ Este método es apropiado para regiones factibles no convexas

Cumplimiento de un número de ecuaciones

- Se deben cumplir al menos *k* de las *N* ecuaciones:

$$
\begin{aligned}
f_1(x_1,\ldots,x_n) &\le 0\\
f_2(x_1,\ldots,x_n) &\le 0\\
&\cdots\\
f_N(x_1,\ldots,x_n) &\le 0
\end{aligned}
$$

- La formulación sería

$$
\begin{aligned}
f_1(x_1,\ldots,x_n) &\le M\delta_1\\
f_2(x_1,\ldots,x_n) &\le M\delta_2\\
&\cdots\\
f_N(x_1,\ldots,x_n) &\le M\delta_N
\end{aligned}
$$

$$
\sum_{i=1}^{N}\delta_i = N-k
$$

$$
\delta_i \in \{0,1\} \quad \forall i = 1,\ldots,N
$$

  - ✓ El caso *N* = 2, *k* = 1 equivale a una disyunción
  - ✓ Se procede de forma análoga para bloques de ecuaciones

Selección entre varios valores

- La función *f* ha de tomar uno de los valores de la siguiente lista:

$$f(x_1,\ldots,x_n)=\begin{cases} d_1 \\ d_2 \\ \vdots \\ d_N \end{cases}$$

- La formulación sería

$$f(x_1,\ldots,x_n)=\sum_{i=1}^{N} d_i\delta_i$$

$$\sum_{i=1}^{N}\delta_i=1$$

$$\delta_i\in\{0,1\}\quad i=1,\ldots,N$$

Implicaciones entre variables binarias y restricciones

$\delta = 1 \Rightarrow \sum_j a_j x_j \le b$	$\sum_j a_j x_j \le b + M(1-\delta)$
$\sum_j a_j x_j \le b \Rightarrow \delta = 1$	$\sum_j a_j x_j \ge b + \varepsilon + (m - \varepsilon)\delta$
$\delta = 1 \Rightarrow \sum_j a_j x_j \ge b$	$\sum_j a_j x_j \ge b + m(1-\delta)$
$\sum_j a_j x_j \ge b \Rightarrow \delta = 1$	$\sum_j a_j x_j \le b - \varepsilon + (M + \varepsilon)\delta$
$\delta = 1 \Rightarrow \sum_j a_j x_j = b$	$\sum_j a_j x_j \le b + M(1-\delta)$ $\sum_j a_j x_j \ge b + m(1-\delta)$
$\sum_j a_j x_j = b \Rightarrow \delta = 1$	$\sum_j a_j x_j \ge b + \varepsilon + (m - \varepsilon)\delta'$ $\sum_j a_j x_j \le b - \varepsilon + (M + \varepsilon)\delta''$ $\delta' + \delta'' - \delta \le 1$

Donde $m \le \sum_j a_j x_j - b \le M$ y $\varepsilon > 0, \varepsilon \simeq 0$

Implicaciones

- La implicación $f(x) \le 0 \Rightarrow g(x) \le 0$ se puede descomponer como $(f(x) \le 0 \Rightarrow \delta = 1)$ y $(\delta = 1 \Rightarrow g(x) \le 0)$ y aplicar las relaciones de la transparencia anterior
- La doble implicación $f(x) \le 0 \Leftrightarrow g(x) \le 0$ se puede descomponer como $(f(x) \le 0 \Leftrightarrow \delta = 1)$ y $(\delta = 1 \Leftrightarrow g(x) \le 0)$ y aplicar las relaciones de la transparencia anterior

Relaciones sencillas entre variables binarias

$\delta_1 = 1 \text{ ó } \delta_2 = 1$	$\delta_1 + \delta_2 \geq 1$
$\delta_1 = 1 \Rightarrow \delta_2 = 1$	$\delta_1 \leq \delta_2$
$\delta_1 = 1 \Leftrightarrow \delta_2 = 1$	$\delta_1 = \delta_2$
$\delta_1 = 1, \ldots, \delta_k = 1 \Rightarrow \delta_1' = 1 \text{ ó} \ldots \text{ó } \delta_l' = 1$	$\delta_1' + \ldots + \delta_l' \geq \delta_1 + \ldots + \delta_k - k + 1$

Al menos k de $\{\delta_j = 1\}_{j=1}^{N}$	$\delta_1 + \ldots + \delta_N \geq k$
Como mucho k de $\{\delta_j = 1\}_{j=1}^{N}$	$\delta_1 + \ldots + \delta_N \leq k$

✓ Si en vez de $\delta = 1$ se tiene $\delta = 0$ en la expresión equivalente se sustituye δ por $1 - \delta$

Ejemplo de fabricación: Formulación 1

- Si se fabrica alguno de los productos A o B entonces debe fabricarse también al menos uno de los productos C, D o E

$$\delta_j = \begin{cases} 1 & \text{si se fabrica el producto } j \\ 0 & \text{si no} \end{cases} \qquad j \in \{A, B, C, D, E\}$$

$$\delta_A = 1 \text{ ó } \delta_B = 1 \Rightarrow \delta_C = 1 \text{ ó } \delta_D = 1 \text{ ó } \delta_E = 1$$

$$\Downarrow$$

$$\begin{cases} \delta_A = 1 \Rightarrow \delta_C = 1 \text{ ó } \delta_D = 1 \text{ ó } \delta_E = 1 \\ \delta_B = 1 \Rightarrow \delta_C = 1 \text{ ó } \delta_D = 1 \text{ ó } \delta_E = 1 \end{cases}$$

$$\Downarrow$$

$$\begin{cases} \delta_C + \delta_D + \delta_E \geq \delta_A \\ \delta_C + \delta_D + \delta_E \geq \delta_B \end{cases}$$

Ejemplo de fabricación: Formulación 2

$$\delta_A = 1 \text{ ó } \delta_B = 1 \Rightarrow \delta_C = 1 \text{ ó } \delta_D = 1 \text{ ó } \delta_E = 1$$

$$\Downarrow$$

$$\delta_A + \delta_B \geq 1 \Rightarrow \delta_C + \delta_D + \delta_E \geq 1$$

$$\Downarrow$$

$$\begin{cases} \delta_A + \delta_B \geq 1 \Rightarrow \delta = 1 \\ \delta = 1 \Rightarrow \delta_C + \delta_D + \delta_E \geq 1 \end{cases}$$

$$\Downarrow$$

$$\begin{cases} 2\delta \geq \delta_A + \delta_B \\ \delta_C + \delta_D + \delta_E \geq \delta \end{cases}$$

- ✓ Hay que añadir las restricciones 0-1 a las variables binarias
- ✓ Si x_j es la cantidad de producto j que se fabrica habría que añadir (en cualquiera de las formulaciones)

$$x_j \leq M\delta_j \ \forall j$$

Ejemplo de baloncesto: Planteamiento

- Un entrenador de baloncesto tiene 9 jugadores, a los que ha evaluado de 1 a 3 de acuerdo con su manejo de pelota, tiro, rebote y defensa, según se indica en la tabla adjunta:

Jugador	Posiciones	Manejo de pelota	Tiro	Rebote	Defensa
1	Pívot	2	1	3	3
2	Base	3	3	1	2
3	Pívot, Alero	2	3	2	2
4	Alero, Base	1	3	3	1
5	Pívot, Alero	1	3	1	2
6	Alero, Base	3	1	2	3
7	Pívot, Alero	3	2	2	1
8	Pívot	2	1	3	2
9	Alero	3	3	1	3

Ejemplo de baloncesto: Planteamiento

- El equipo titular de 5 jugadores debe tener la máxima capacidad defensiva y satisfacer las siguientes condiciones:
  - ✓ Al menos dos jugadores deben estar en disposición de actuar de pívot, dos de alero y uno de base
  - ✓ Su nivel medio en el manejo de pelota en tiro y en rebote debe ser por lo menos 2
  - ✓ Si juega el jugador 3, entonces el jugador 6 no puede jugar
  - ✓ Si juega el jugador 1, también deberá jugar el 4 ó el 5, pero no los dos a la vez
  - ✓ El jugador 8 ó el 9, pero no los dos a la vez, deben jugar
- Formular un programa lineal que facilite la selección del equipo titular

Ejemplo de baloncesto: Formulación

- Variables:

$$x_j = \begin{cases} 1 & \text{si se incluye el jugador } j \text{ en el equipo} \\ 0 & \text{en otro caso} \end{cases}$$

$$j = 1, \ldots, 9$$

$$x_{jk} = \begin{cases} 1 & \text{si se incluye el jugador } j \text{ en posición } k \\ 0 & \text{en otro caso} \end{cases}$$

$$j = 3, 4, 5, 6, 7$$

$$k = p, a, b \text{ (sólo los necesarios)}$$

- Función objetivo:

$$\max 3x_1 + 2x_2 + 2x_3 + x_4 + 2x_5 + 3x_6 + x_7 + 2x_8 + 3x_9$$

- Restricciones:

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 = 5$$ 5 jugadores en el equipo

$$x_1 + x_{3p} + x_{5p} + x_{7p} + x_8 \geq 2$$ Al menos 2 pívots

$$x_{3a} + x_{4a} + x_{5a} + x_{6a} + x_{7a} + x_9 \geq 2$$ Al menos 2 aleros

$$x_2 + x_{4b} + x_{6b} \geq 1$$ Al menos 1 base

Ejemplo de baloncesto: Formulación

$$\begin{cases} 2x_1 + 3x_2 + 2x_3 + x_4 + x_5 + 3x_6 + 3x_7 + 2x_8 + 3x_9 \geq 10 \\ x_1 + 3x_2 + 3x_3 + 3x_4 + 3x_5 + x_6 + 2x_7 + x_8 + 3x_9 \geq 10 \\ 3x_1 + x_2 + 2x_3 + 3x_4 + x_5 + 2x_6 + 2x_7 + 3x_8 + x_9 \geq 10 \end{cases}$$

Nivel medio al menos 2 en manejo, tiro y rebote

$$x_3 + x_6 \leq 1$$

Equivale a $x_3 = 1 \Rightarrow x_6 = 0$

$$\begin{cases} x_4 + x_5 \leq 2 - x_1 \\ x_4 + x_5 \geq x_1 \end{cases}$$

Equivalen a $x_1 = 1 \Rightarrow x_4 + x_5 = 1$

$$x_8 + x_9 = 1$$

Deben jugar 8 ó 9, pero no ambos

$$\begin{cases} x_{3p} + x_{3a} - x_3 = 0 \\ x_{4a} + x_{4b} - x_4 = 0 \\ x_{5p} + x_{5a} - x_5 = 0 \\ x_{6a} + x_{6b} - x_6 = 0 \\ x_{7p} + x_{7a} - x_7 = 0 \end{cases}$$

Coherencia entre puestos para los jugadores polivalentes

$$x_j, x_{jk} \in \{0,1\} \; \forall j,k$$

Productos

$\delta_1\delta_2 = 0$ $\delta_i \in \{0,1\}$	$\delta_1 = 0$ ó $\delta_2 = 0$	$\delta_1 + \delta_2 \le 1$ $\delta_i \in \{0,1\}$
$\delta_1\delta_2$ $\delta_i \in \{0,1\}$	Reemplazar $\delta_1\delta_2$ por δ_3 $\delta_3 = 1 \Leftrightarrow \delta_1 = 1 \text{ y } \delta_2 = 1$	$\delta_3 \le \delta_1$ $\delta_3 \le \delta_2$ $\delta_3 \ge \delta_1 + \delta_2 - 1$ $\delta_i \in \{0,1\}$
$x\delta$ $x \ge 0$ $\delta \in \{0,1\}$	Reemplazar $x\delta$ por y $\delta = 0 \Rightarrow y = 0$ $\delta = 1 \Rightarrow y = x$	$y \ge 0$ $y \le M\delta$ $y \le x$ $y \ge x + M\delta - M$ $(\text{donde } x \le M)$

Modelado de funciones objetivo no lineales

- ❑ Problemas con coste fijo (fixed-charge)
- ❑ Problemas con costes variables por tramos
- ❑ Función objetivo minimax o maximin

Problemas con coste fijo

- Cada variable x_j lleva asociado un coste fijo k_j y un coste unitario c_j:

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

✓ Se introduce una variable binaria y_j por cada x_j que cumpla

$$y_j = \begin{cases} 1 & x_j > 0 \\ 0 & x_j = 0 \end{cases}$$

✓ Para ello se añaden restricciones de la forma

$$x_j \leq M \cdot y_j \qquad y_j \in \{0,1\} \qquad x_j \geq 0 \qquad (M \text{ “grande”})$$

✓ La función objetivo queda

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

Problemas con costes variables por tramos

- La variable *x* lleva asociados costes c^1, c^2,…, c^N en los tramos $(p^0, p^1]$, $(p^1, p^2]$,…, $(p^{N-1}, p^N]$
  - ✓ Se introducen *N* variables continuas y *N* variables binarias de la siguiente forma

$$x^k = \begin{cases} x & \text{si } x \in \left(p^{k-1}, p^k\right] \\ 0 & \text{si no} \end{cases} \qquad \delta^k = \begin{cases} 1 & \text{si } x^k > 0 \\ 0 & x^k = 0 \end{cases}$$

  - ✓ En la función objetivo se introduce el sumando

$$cx = \sum_{k=1}^{N} c^k x^k$$

  - ✓ Se añaden las restricciones

$$\sum_{k=1}^{N} x^k = x, \quad \sum_{k=1}^{N} \delta^k = 1$$

$$(p^{k-1} + \varepsilon)\delta^k \le x^k \le p^k \delta^k \qquad \forall k = 1,\dots,N$$

$$\delta^k \in \left\{0,1\right\} \qquad \forall k = 1,\dots,N$$

Función objetivo minimax o maximin

$$\min_x f(x)$$
$$Ax = b$$
$$x \geq 0$$
$$x \in \mathbb{R}^n, A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$$

siendo

$$f(x) = \max\left\{c_1^T x + d_1, c_2^T x + d_2, \ldots, c_p^T x + d_p\right\}$$
$$c_1, c_2, \ldots, c_p \in \mathbb{R}^n, \quad d_1, d_2, \ldots, d_p \in \mathbb{R}$$

- El problema se puede reformular como

$$\boxed{\begin{aligned}&\min z\\ &z \geq c_1^T x + d_1\\ &z \geq c_2^T x + d_2\\ &\ldots\\ &z \geq c_p^T x + d_p\\ &Ax = b\\ &x \geq 0\end{aligned}}$$

✓ Análogo para función objetivo maximin