ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA

DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

Modelos matemáticos de optimización

Andrés Ramos

Andres.Ramos@iit.icai.upcomillas.es

Universidad Pontificia Comillas

Begoña Vitoriano

bvitoriano@mat.ucm.es

Universidad Complutense de Madrid

Índice

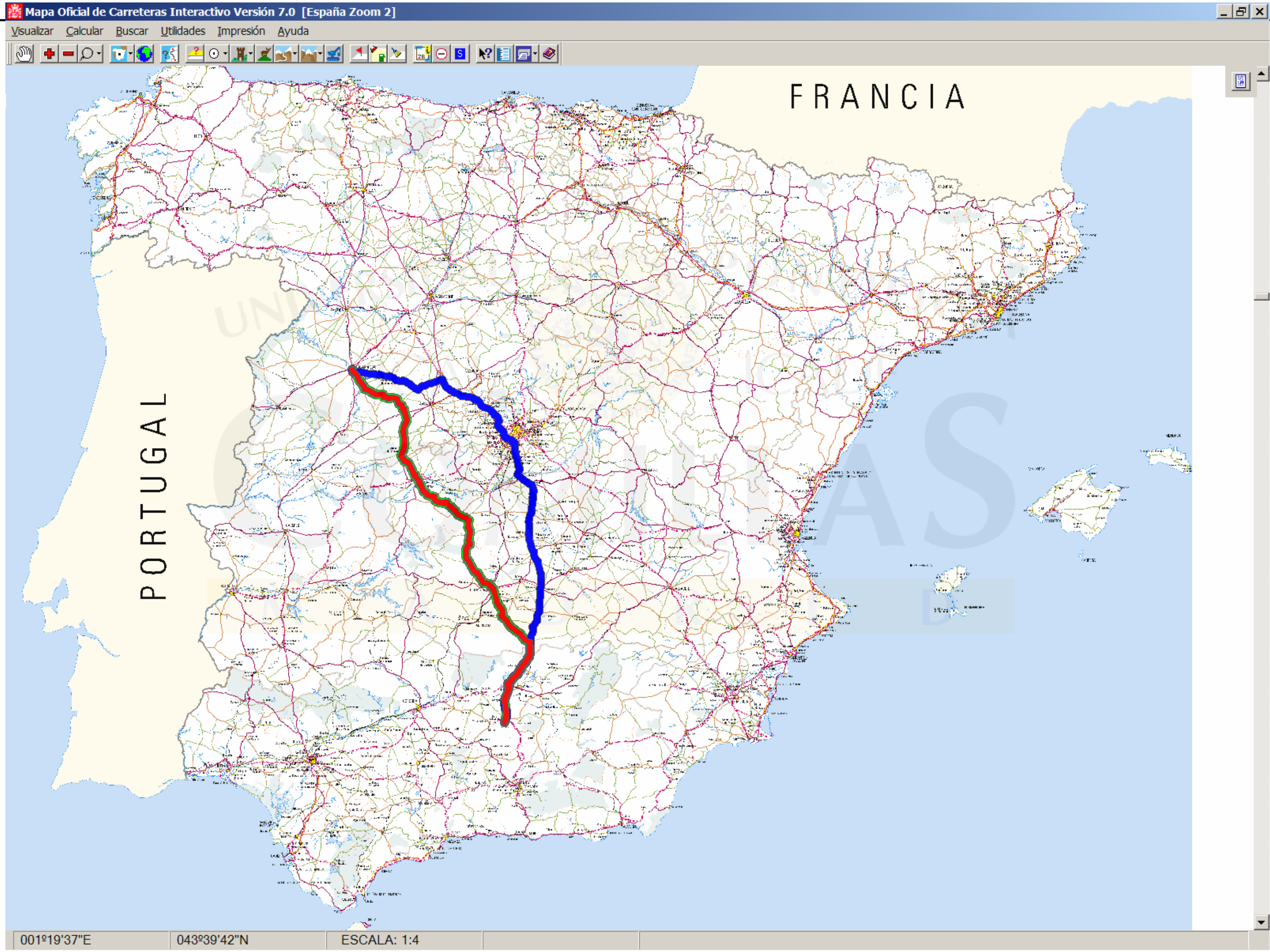
- ➢ Optimización

- ❑ Modelo y modelado
- ❑ Etapas en el desarrollo de un modelo
- ❑ Problemas característicos
- ❑ Modelado de implicaciones lógicas
- ❑ Desarrollo de modelos de optimización

Definición de la Investigación Operativa (IO)

- Aplicación de métodos científicos en la mejora de la efectividad en las operaciones, decisiones y gestión.
  - diseño y mejora de las operaciones y decisiones
  - resolución de problemas y ayuda en las funciones de gestión, planificación o predicción
  - aportan conocimiento y ayuda en la toma de decisiones
- Tareas:
  - recoger y analizar datos
  - desarrollar y probar modelos matemáticos
  - proponer soluciones o recomendaciones
  - interpretar la información
  - ayudar a implantar acciones de mejora
- Resultados: aplicaciones informáticas, sistemas, servicios o productos.

Problema de camino mínimo



¿Qué es la optimización?

- Optimización: encontrar una alternativa de decisión con la propiedad de ser mejor que cualquier otra en algún sentido
- Componentes
  - ✓ *Función objetivo*
    - Medida cuantitativa del funcionamiento (de la bondad) de un sistema que se desea maximizar o minimizar
  - ✓ *Variables*
    - Decisiones que afectan el valor de la función objetivo
    - Independientes o dependientes
  - ✓ *Restricciones*
    - Conjunto de relaciones que las variables están obligadas a satisfacer
- Resolver: encontrar el valor que deben tomar las *variables* para hacer óptima la *función objetivo* satisfaciendo el conjunto de *restricciones*.

Optimización clásica vs. Metaheurística (i)

Métodos clásicos	•programación lineal •programación lineal entera mixta •programación cuadrática •programación no lineal •optimización estocástica •programación dinámica •teoría de grafos u optimización en redes
Métodos metaheurísticos (Inteligencia Artificial)	•algoritmos evolutivos (genéticos) •recocido o templado simulado (*simulated annealing*) •búsquedas tabú, aleatoria, avariciosa •sistemas multiagente (colonias de hormigas)

Optimización clásica vs. Metaheurística (ii)

- Métodos clásicos
  - ✓ buscan el óptimo “localmente”
  - ✓ garantizan el óptimo numérico
  - ✓ permiten un elevado número de restricciones
- Métodos metaheurísticos
  - ✓ Imitan fenómenos sencillos observados en la naturaleza
  - ✓ “globales”, mecanismos específicos para evitar óptimos locales
  - ✓ NO garantizan la obtención del óptimo
  - ✓ NO permiten elevado número de restricciones
  - ✓ exploran gran número de soluciones en tiempo muy corto
  - ✓ aplicados principalmente a problemas combinatoriales

Índice

- ❑ Optimización
- ➢ Modelo y modelado
- ❑ Etapas en el desarrollo de un modelo
- ❑ Problemas característicos
- ❑ Modelado de implicaciones lógicas
- ❑ Desarrollo de modelos de optimización

Modelo

Modelo

- Definición
  - ✓ Esquema teórico, generalmente en forma matemática, de un sistema o de una realidad compleja (por ejemplo, la evolución económica de un país), que se elabora para facilitar su comprensión y el estudio de su comportamiento. (Diccionario de la lengua española. Real Academia Española)
- Representación precisa de una realidad
- Herramienta de ayuda a la toma de decisiones
- Puede involucrar equipo multidisciplinar
- Equilibrio entre representación detallada y capacidad de obtener la solución
- *Modelador*: especifica y desarrolla el modelo
- *Experto*: conoce el problema real

Dos riesgos importantes

- Modelado exhaustivo, cuasi real. Puede ocasionar la carencia de un algoritmo que solucione el problema
- Modelado simplista para utilizar un algoritmo disponible. Pueden llegar a darse soluciones de un problema que no existe

- El modelado debe ser un compromiso entre ambos casos patológicos

Modelado

- Ciencia
  - ✓ Análisis y detección de las relaciones entre datos
  - ✓ Suposiciones y aproximaciones a los problemas
  - ✓ Algoritmos específicos de solución
  - ✓ Soluciones del modelo
- Arte
  - ✓ Visión o interpretación de la realidad
  - ✓ Estilo en modelo y documentación
  - ✓ Elegancia y simplicidad en desarrollo
  - ✓ Uso de creativo de herramientas

Beneficios del modelado

- ❑ Diálogo entre modelador y experto
- ❑ Organiza información disponible
- ❑ Estructura la compresión del comportamiento del sistema
- ❑ Internaliza estructura organizativa de empresa
- ❑ Permite compartir supuestos entre modelador y experto
- ❑ Proporciona una herramienta para el análisis
- ❑ Indica dirección de mejora en decisiones

Índice

- Optimización
- Modelo y modelado

- Etapas en el desarrollo de un modelo

- Problemas característicos
- Modelado de implicaciones lógicas
- Desarrollo de modelos de optimización

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

Identificación del problema

- Recolección de información relevante
- Definición del problema en términos vagos
- Interpretación y traducción a términos precisos
- Datos son vitales, suelen ser cuello de botella
- Etapa fundamental para que decisiones sean útiles

Es imprescindible asegurarse de que el modelo representa adecuadamente la realidad que pretende reflejar.

Datos de entrada

- GIGOLO
- Garbage In, Garbage Out, Look Out !

El mejor modelo no sirve de nada si los datos de entrada no están adecuadamente refinados

Especificación matemática y formulación

- Definición de variables, ecuaciones, función objetivo, parámetros
- Identificación de tipo de problema (LP, MIP, NLP)
- Énfasis en precisión y belleza en la formulación
- Análisis de tamaño y estructura del problema
- Categorías de problemas LP según su tamaño

	Restricciones	Variables
✓ Caso ejemplo	100	100
✓ Tamaño medio	10000	10000
✓ Gran tamaño	500000	500000
✓ Muy gran tamaño	> 500000	> 500000

Diseño conceptual

- Al que tiene un martillo todo le parecen clavos.

La técnica de modelado, así como el nivel de detalle del mismo y en general cualquier aspecto de modelado debe adecuarse a la estructura del problema, a las expectativas y necesidades del cliente…

Modelo conceptual

- Un mes de programación puede “ahorrarte” un par de horas de biblioteca.

El tiempo de menos dedicado al modelo conceptual retrasa de forma exponencial la implantación del modelo.

Resolución

- Algoritmo de obtención de solución óptima, cuasióptima o, al menos, satisfactoria
- Detección de soluciones cuasióptimas atractivas
- Diferentes métodos de solución
- Diferentes implantaciones del algoritmo elegido

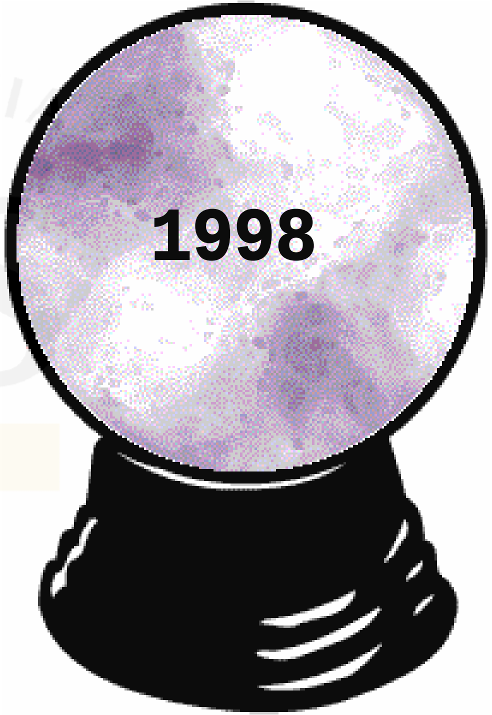
Verificación, validación y refinamiento

- Eliminación de errores en codificación
- Comprobar validez de simplificaciones adoptadas
- Comprobación de adaptación a la realidad
- Ampliación en el modelado por nuevas necesidades

Modelo: validación

- El pasado no es estocástico

El contraste de los resultados del modelo con datos reales del sistema es imprescindible.



Interpretación y análisis de resultados

- Análisis de sensibilidad en parámetros de entrada
- Robustez de la solución óptima

Implantación, documentación y mantenimiento

- Etapa fundamental para el éxito de un modelo
- Documentación clara, precisa y completa
- Manual de usuario con especificación técnica funcional, matemática e informática
- Formación de posibles usuarios

Índice

- ❑ Optimización
- ❑ Modelo y modelado
- ❑ Etapas en el desarrollo de un modelo

- ➢ Problemas característicos

- ❑ Modelado de implicaciones lógicas
- ❑ Desarrollo de modelos de optimización

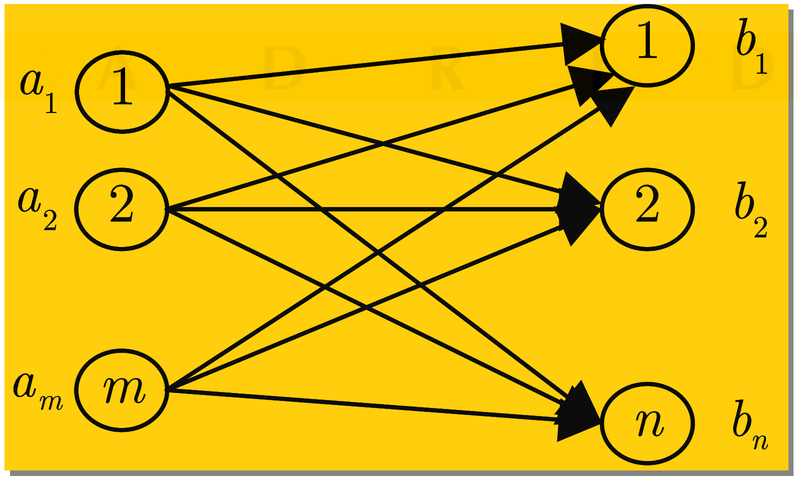
Algunos problemas característicos de LP y BIP

- Se han estudiado exhaustivamente. Su importancia práctica es limitada, pero pueden formar parte de otros problemas.
- *Programación lineal LP*
  - ✓ Transporte
  - ✓ Transbordo
  - ✓ Asignación
- *Programación binaria pura BIP*
  - ✓ Mochila
  - ✓ Recubrimiento
  - ✓ Empaquetado
  - ✓ Partición
  - ✓ Viajante

Problema de transporte

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

- a_i oferta de producto en el origen i — m orígenes
- b_j demanda de producto en el destino j — n destinos
- c_{ij} coste unitario de transporte desde i a j



Formulación problema de transporte

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

- Oferta disponible en cada origen i $\quad \sum_{j=1}^{n} x_{ij} = a_i \quad \forall i = 1, \ldots, m$
- Demanda de cada destino j $\quad \sum_{i=1}^{m} x_{ij} = b_j \quad \forall j = 1, \ldots, n$
- $x_{ij} \geq 0$ unidades de producto transportadas desde i hasta j $\forall i, j$
- Se supone que la oferta es igual a la demanda del producto

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

- 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 muy elevado

Estructura problema de transporte

$$
\begin{array}{lccccccccccccc}
 & x_{11} & x_{12} & \cdots & x_{1n} & x_{21} & x_{22} & \cdots & x_{2n} & \cdots & x_{m1} & x_{m2} & \cdots & x_{mn} \\
 & 1 & 1 & \cdots & 1 & & & & & & & & & \\
m \text{ restricciones de oferta} & & & & & 1 & 1 & \cdots & 1 & & & & & \\
 & & & & & & & & & \ddots & & & & \\
 & & & & & & & & & & 1 & 1 & \cdots & 1 \\
 & 1 & & & & 1 & & & & \cdots & 1 & & & \\
n \text{ restricciones de demanda} & & 1 & & & & 1 & & & \cdots & & 1 & & \\
 & & & \ddots & & & & \ddots & & \cdots & & & \ddots & \\
 & & & & 1 & & & & 1 & \cdots & & & & 1
\end{array}
$$

Si a_i y b_j son enteros $\Rightarrow$ x_{ij} son enteros por ser la matriz totalmente unimodular (i.e., toda submatriz cuadrada tiene determinante 0, 1 ó –1)

Problema de trasbordo

- Determinar en una red con n nodos las rutas más baratas para llevar unidades de un producto desde sus orígenes a sus destinos pasando por puntos de trasbordo intermedios.

- Cada *origen* genera $b_i > 0$ unidades.
- Cada *destino* consume $b_i < 0$ unidades.
- Cada *trasbordo* ni genera ni consume unidades $b_i = 0$.
- c_{ij} coste unitario de transporte desde i hasta j en dicho sentido.

Formulación problema de trasbordo

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

- Balance o conservación del flujo en cada nudo *i*

$$\sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = b_i \quad \forall i = 1, \ldots, n$$

- $x_{ij} \geq 0$ unidades de producto transportadas desde *i* a *j* $\forall i, j$
- Se supone que la oferta es igual a la demanda del producto

$$\sum_{i=1}^{n} b_i = 0$$

Problema de asignación de tareas

- n tareas

  n personas (máquinas, etc.) para realizarlas
- Es un caso particular del problema de transporte.
- Minimizar el coste total de realizar las tareas sabiendo que cada persona realiza 1 tarea y cada tarea es realizada por 1 persona.
- c_{ij} coste de realizar la tarea i por la persona j

$$x_{ij}\begin{cases}1 & \text{si la tarea } i \text{ es realizada por la persona } j\\ 0 & \text{en cualquier otro caso}\end{cases}$$

- *Aunque no es necesario declararlas como binarias.*

Formulación problema de asignación de tareas

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

- Cada tarea i es hecha por una persona

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

- Cada persona j realiza una tarea

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

$$x_{ij} \geq 0 \quad \forall i, j$$

Problema de la mochila (*knapsack*)

- n proyectos
- Maximizar el valor total de la elección de un conjunto de proyectos sin sobrepasar el presupuesto disponible.
- c_j coste de cada proyecto j
- v_j valor de cada proyecto j
- b presupuesto disponible

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

Formulación problema de la mochila

$$\max_{x_j} \sum_{j=1}^{n} v_j x_j$$

- Limitación del presupuesto disponible

$$\sum_{j=1}^{n} c_j x_j \le b$$

$$x_j \in \{0,1\} \quad \forall j$$

Problema de recubrimiento (*set covering*)

- m características (vuelos)
- n combinación de características (secuencia de vuelos). La elección de una combinación implica realizar todas las características de la misma.
- Minimizar el coste total de las combinaciones elegidas de manera que se cubra (posea) cada característica *al menos* una vez.
- c_j coste de elegir la combinación j
- matriz de pertenencia $a_{ij} \begin{cases} 1 & \text{si la característica } i \text{ pertenece a la combinación } j \\ 0 & \text{si no pertenece} \end{cases}$

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

Formulación problema de recubrimiento

$$\min_{x_j} \sum_{j=1}^{n} c_j x_j$$

- Cada característica *i* del conjunto de todas las combinaciones *j* que la poseen debe ser escogida al menos una vez.

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1 \quad i = 1, \ldots, m$$

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

Ejemplo de recubrimiento: asignación de tripulaciones

- Una compañía aérea necesita asignar sus tripulaciones para cubrir todos sus vuelos. En particular, quiere resolver el problema de asignar TRES tripulaciones con base en San Francisco a los vuelos listados en la primera columna de la tabla. Las otras columnas muestran las 12 SECUENCIAS FACTIBLES de vuelos para una tripulación cualesquiera. Los números de cada columna indican el orden de los vuelos. Se necesita elegir tres secuencias (una por tripulación) de manera que se cubran todos los vuelos. Se permite tener más de una tripulación en un vuelo, donde la/s tripulación/es extra viajan como pasajeros, pero por convenio laboral la tripulación extra cobra como si estuviera trabajando. El coste de asignación de una tripulación a cada secuencia de vuelos se da en millones de euros en la última fila. El objetivo es minimizar el coste total de asignación de las tres tripulaciones para cubrir todos los vuelos.
- Resolver el mismo problema para el caso en que no se permite el vuelo de una tripulación fuera de servicio en un vuelo.

Secuencias factibles de vuelo

	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

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

- Cobertura de cada vuelo

$$\begin{aligned} &x_1 + x_4 + x_7 + x_{10} \geq 1 \\ &x_2 + x_5 + x_8 + x_{11} \geq 1 \\ &x_3 + x_6 + x_9 + x_{12} \geq 1 \\ &\vdots \end{aligned}$$

- Asignación de las tres tripulaciones

$$\sum_{j=1}^{12} x_j = 3$$

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

$$x_j = \begin{cases} 1 & \text{si se elige la secuencia } j \text{ para una tripulación} \\ 0 & \text{en cualquier otro caso} \end{cases}$$

- Solución
  - ✓ $x_3 = x_4 = x_{11} = 1$ $\quad x_j = 0 \quad j \neq 3, 4, 11$ $\quad$ coste = 18 M€
  - ✓ $x_1 = x_5 = x_{12} = 1$ $\quad x_j = 0 \quad j \neq 1, 5, 12$ $\quad$ coste = 18 M€

Problema de empaquetado (*set packing*)

- m proyectos
- n paquetes (conjuntos) de proyectos. La elección de un paquete (conjunto) implica realizar todos los proyectos del mismo.
- Maximizar el beneficio total de manera que ningún proyecto se realice más de una vez.
- c_j beneficio de elegir el paquete j

$$a_{ij}\begin{cases}1 & \text{si el proyecto } i \text{ está en el paquete } j\\ 0 & \text{si no lo está}\end{cases}$$

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

Formulación problema de empaquetado

$$\max_{x_j} \sum_{j=1}^{n} c_j x_j$$

- Cada proyecto *i* del conjunto de todos los paquetes que lo incluyen no puede ser elegido más de una vez

$$\sum_{j=1}^{n} a_{ij} x_j \leq 1 \quad i = 1, \ldots, m$$

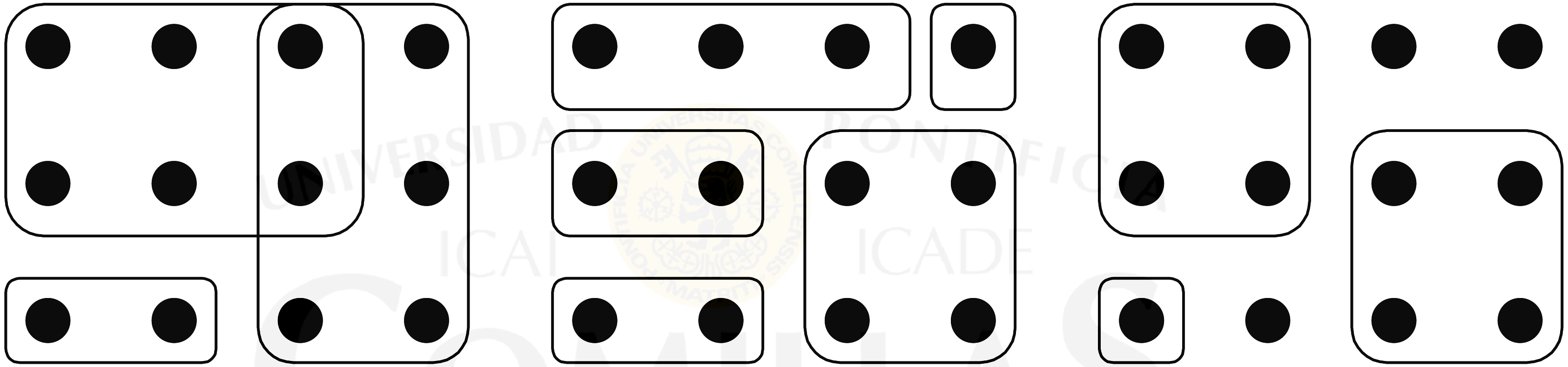
$$x_j \in \{0,1\} \quad j = 1, \ldots, n$$

Problema de partición (*set partitioning*)

- EXACTAMENTE una característica (proyecto) del conjunto de combinaciones (paquetes) que la contienen debe ser elegida

$$\sum_{j=1}^{n} a_{ij} x_j = 1 \quad i = 1, \ldots, m$$

Problemas de recubrimiento, partición y empaquetado



RECUBRIMIENTO PARTICIÓN EMPAQUETADO

Problema del viajante (*traveling salesman problem TSP*)

- Consiste en hacer un recorrido que pase por ciudades sin repetir ninguna y volviendo a la ciudad de partida de manera que la distancia total sea mínima.
- *Formulación 1:*

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

$$\min_{x_{ij}} \sum_i \sum_j c_{ij} x_{ij}$$

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

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

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

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

Problema del viajante (TSP)

- *Formulación 2:*

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

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

$$\sum_{j,k} x_{ijk} = 1 \quad \forall i$$

$$\sum_{i,k} x_{ijk} = 1 \quad \forall j$$

$$\sum_{i,j} x_{ijk} = 1 \quad \forall k$$

$$\sum_{i} x_{ijk} = \sum_{r} x_{jrk+1} \quad \forall j,k$$

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

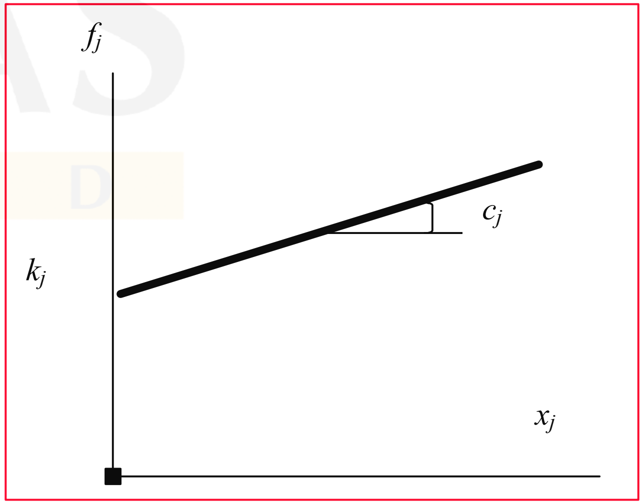
Problema de coste fijo

- Se tiene la función objetivo $f_j(x_j) = \begin{cases} 0 & x_j = 0 \\ k_j + c_j x_j & x_j > 0 \end{cases}$
- Definimos una variable binaria que modela la decisión binaria sobre la realización de la actividad x_j

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

- La formulación resultante es

$$\min \sum_{j=1}^{n} f_j(x_j) = \sum_{j=1}^{n} \left(k_j y_j + c_j x_j \right)$$
$$x_j \le M y_j$$
$$x_j \ge 0 \quad j = 1,\dots,n$$
$$y_j \in \{0,1\} \quad j = 1,\dots,n$$



Asignación de grupos térmicos

- ¿Qué grupos térmicos de generación eléctrica hay que acoplar en cada hora del día (o semana) de manera que:

  - ✓ Se minimicen los costes variables de generación (incluyendo costes de combustible y costes de arranque y parada)
  - ✓ Se suministre la demanda en cada hora
  - ✓ Se mantenga un cierto nivel de reserva rodante
  - ✓ Se respeten los parámetros de funcionamiento de los grupos térmicos (mínimos técnicos, rampas de subida y bajada)

Datos y variables

DATOS

D_h demanda térmica en la hora *h* [MW]

R coeficiente de reserva rodante con respecto a la demanda [p.u.]

a_t término lineal del coste de combustible del grupo térmico *t* [€/MWh]

b_t término fijo del coste de combustible del grupo térmico t [€/h]

ca_t coste de arranque del grupo térmico *t* [€]

cp_t coste de parada del grupo térmico *t* [€]

$\overline{P}_t$ potencia máxima del grupo térmico *t* [MW]

$\underline{P}_t$ potencia mínima del grupo térmico *t* [MW]

rs_t rampa de subida del grupo térmico *t* [MW/h]

rb_t rampa de bajada del grupo térmico *t* [MW/h]

VARIABLES

P_{ht} potencia producida por el grupo térmico *t* en la hora *h* [MW]

A_{ht} acoplamiento del grupo térmico *t* en la hora *h* [0,1]

AR_{ht} arranque del grupo térmico *t* en la hora *h* [0,1]

PR_{ht} parada del grupo térmico *t* en la hora *h* [0,1]

Formulación

$$\min \sum_{h=1}^{H}\sum_{t=1}^{T}\left(a_t P_{ht} + b_t A_{ht} + ca_t AR_{ht} + cp_t PR_{ht}\right)$$

$$\sum_{t=1}^{T} P_{ht} = D_h \qquad H$$

$$\sum_{t=1}^{T} (\bar{P}_t A_{ht} - P_{ht}) = RD_h \qquad H$$

$$\begin{aligned}
&\underline{P}_t A_{ht} \le P_{ht} \le \bar{P}_t A_{ht} && 2HT \\
&A_{ht} - A_{h-1t} = AR_{ht} - PR_{ht} && (H-1)T \\
&P_{ht} - P_{h-1t} \le rs_t && (H-1)T \\
&P_{h-1t} - P_{ht} \le rb_t && (H-1)T
\end{aligned}$$

$$P_{ht} \ge 0 \qquad A_{ht}, AR_{ht}, PR_{ht} \in \{0,1\}$$

Problemas de programación no lineal (i)

- Problema de transporte con descuentos por cantidad
  - ✓ El precio unitario de transporte entre un origen y un destino es decreciente en función de la cantidad a transportar.
- Problema de flujo de cargas en un sistema eléctrico
  - ✓ Las pérdidas son no lineales
- Problema de producción con elasticidad en el precio y/o en el coste
  - ✓ Función de la demanda o curva precio-demanda $p(x)$ representa el precio unitario que se necesita para poder vender x unidades. Es una función decreciente, nunca inferior al coste unitario de producción c. Los ingresos brutos (producto de cantidad producida por precio) es una expresión no lineal. Margen de contribución (beneficio bruto, EBITDA) $$P(x) = xp(x) - cx$$
  - ✓ Los costes no lineales pueden aparecer por una mayor eficiencia unitaria en función de la cantidad.

Problemas de programación no lineal (ii)

- Problema de selección de una cartera de inversiones

  n tipos de acciones

  x_j, j=1,…,n representan el número de acciones j que se van a incluir en la cartera

  μ_j y σ_{jj} la media y la varianza del rendimiento sobre cada acción de tipo j, en donde σ_{jj} es una medida del riesgo de estas acciones. Sea σ_{ij} la covarianza del rendimiento sobre una acción de cada tipo i y j.

  $R(x)$ rendimiento esperado y su varianza $V(x)$

$$R(x) = \sum_{j=1}^{n} \mu_j x_j$$

$$V(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j$$

  la función objetivo es $f(x) = R(x) - \beta V(x)$

  siendo β el factor de aversión al riesgo.

Índice

- ❑ Optimización
- ❑ Modelo y modelado
- ❑ Etapas en el desarrollo de un modelo
- ❑ Problemas característicos

➢ Modelado de implicaciones lógicas

- ❑ Desarrollo de modelos de optimización

Modelado de implicaciones

- Queremos modelar la condición de que "*si se produce el producto A también se debe producir el producto B*". La condición de producción de un producto *j* la representamos por la restricción $x_j \geq 1$. Luego esta implicación es $x_A \geq 1 \Rightarrow x_B \geq 1$

- Esta condición no se puede introducir directamente en un problema lineal porque hace que la estructura del problema (el que se considere o no una restricción más $x_B \geq 1$) depende de que se cumpla otra ($x_A \geq 1$) y esto sólo se conoce una vez que se ha determinado la solución óptima. Un problema de optimización no se puede redefinir endógenamente, es decir, en función de los propios valores que toman las variables del problema.

Restricciones disyuntivas (i)

- Pareja de restricciones donde sólo una (cualquiera de las dos) debe satisfacerse, mientras que la otra no es necesario que se cumpla. Debe cumplirse una pero no necesariamente las dos.

$$f(x) \leq 0 \quad \text{ó} \quad g(x) \leq 0$$

Restricciones disyuntivas (ii)

- Queremos cumplir una de estas dos restricciones

$$3x_1 + 2x_2 \leq 18 \quad \text{ó} \quad x_1 + 4x_2 \leq 16$$

- Añadir *M* (constante de valor elevado) equivale a relajar la restricción (para variables positivas con coeficientes positivos)
  - ✓ Relajo la restricción 1 y satisfago la 2

$$\begin{aligned} 3x_1 + 2x_2 &\leq 18 + M \\ x_1 + 4x_2 &\leq 16 \end{aligned}$$

  - ✓ Relajo la restricción 2 y satisfago la 1

$$\begin{aligned} 3x_1 + 2x_2 &\leq 18 \\ x_1 + 4x_2 &\leq 16 + M \end{aligned}$$

- Mediante variable binaria auxiliar elijo cuál de las dos relajo

$$\begin{aligned} 3x_1 + 2x_2 &\leq 18 + M\delta \\ x_1 + 4x_2 &\leq 16 + M(1-\delta) \end{aligned}$$

$$\delta = \begin{cases} 1 & \text{se relaja la ecuación 1} \\ 0 & \text{se relaja la ecuación 2} \end{cases}$$

Cumplir al menos *k* de *N* ecuaciones

- Se tienen que cumplir al menos k de N ($k < N$) ecuaciones

$$
\begin{aligned}
& f_1(x_1,\ldots,x_n) \le d_1 \\
& f_2(x_1,\ldots,x_n) \le d_2 \\
& \vdots \\
& f_N(x_1,\ldots,x_n) \le d_N
\end{aligned}
$$

- $k = 1$ y $N = 2$ es el caso anterior
- Formulación

$$
\begin{aligned}
& f_1(x_1,\ldots,x_n) \le d_1 + M\delta_1 \\
& f_2(x_1,\ldots,x_n) \le d_2 + M\delta_2 \\
& \vdots \\
& f_N(x_1,\ldots,x_n) \le d_N + M\delta_N \\
& \sum_{i=1}^{N} \delta_i = N - k \\
& \delta_i \in \{0,1\} \quad i = 1,\ldots,N
\end{aligned}
$$

Seleccionar uno entre *N* valores

- La ecuación se debe cumplir para exactamente uno de los valores

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

- Formulación

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

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

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

Implicaciones sencillas

- Retomemos el ejemplo de la restricción que aparecía en el problema de coste fijo $x \le M\delta$

  siendo M una cota superior positiva de x y δ la variable binaria.
  - ✓ Si $\delta = 1$ la restricción no obliga a nada ya que $x \le M$ se cumple por definición.
  - ✓ Si $\delta = 0$ entonces $x \le 0$.
- Luego esta restricción permite modelar la implicación $\delta = 0 \Rightarrow x \le 0$
- Por otra parte, si $x > 0$ entonces $\delta = 1$. Si $x \le 0$ la restricción no obliga a nada. $x > 0 \Rightarrow \delta = 1$
- Ambas son implicaciones equivalentes puesto que $P \to Q$ es equivalente a $\text{No } Q \to \text{No } P$

$$\left.\begin{array}{l} \delta = 0 \Rightarrow x \le 0 \\ x > 0 \Rightarrow \delta = 1 \end{array}\right\} x \le M\delta$$

Implicaciones sencillas (ii)

- De forma análoga veamos la restricción $x \geq m\delta$ siendo m una cota inferior negativa de x y δ la variable binaria.
  - ✓ Si $\delta = 1$ la restricción no obliga a nada ya que $x \geq m$ se cumple por definición.
  - ✓ Si $\delta = 0$ entonces $x \geq 0$. Luego esta restricción permite modelar la implicación $\delta = 0 \Rightarrow x \geq 0$
- Por otra parte, si $x < 0$ entonces $\delta = 1$. Si $x \geq 0$ la restricción no obliga a nada. $x < 0 \Rightarrow \delta = 1$
- Nuevamente ambas son implicaciones equivalentes puesto que $P \rightarrow Q$ es equivalente a $\text{No } Q \rightarrow \text{No } P$

$$\left.\begin{array}{l} \delta = 0 \Rightarrow x \geq 0 \\ x < 0 \Rightarrow \delta = 1 \end{array}\right\} x \geq m\delta$$

Índice

- ❑ Optimización
- ❑ Modelo y modelado
- ❑ Etapas en el desarrollo de un modelo
- ❑ Problemas característicos
- ❑ Modelado de implicaciones lógicas

➢ Desarrollo de modelos de optimización

Alternativas desarrollo modelos optimización

- Lenguajes de programación de propósito general (C, C++, Java, Visual Basic, FORTRAN 90)
  - ✓ C (CPLEX de ILOG, OSL de IBM)
  - ✓ C++ (Concert de ILOG, LINDO API de LINDO Systems, OptiMax 2000 de Maximal Software, FLOPC++ de Universidade de Aveiro)
  - ✓ Dominio público [GNU Linear Programming Toolkit GLPK (www.gnu.org/software/glpk), Computational Infrastructure for Operations Research COIN-OR (www.coin-or.org), LP solver SoPlex (http://soplex.zib.de) and MIP framework SCIP (http://scip.zib.de)]
- Lenguajes o entornos de cálculo numérico o simbólico (hojas de cálculo, Matlab, Mathematica)
- Lenguajes algebraicos de modelado [GAMS, OPL Studio, AMPL, AIMMS, XPRESS-MP, MPL, Zimpl (http://zimpl.zib.de) (el último de dominio público)]

- En *OR/MS Today* (www.orms-today.com) una vez al año hay artículos de resumen de los diferentes entornos de optimización y sus características

Optimizadores en hojas de cálculo

- Ventajas
  - ✓ Fáciles de usar
  - ✓ Integración total con la hoja de cálculo
  - ✓ Familiaridad con el entorno que facilita la explicación del modelo y de sus resultados
  - ✓ Facilidad de presentación de resultados en gráficos
- Inconvenientes
  - ✓ No inducen una buena práctica de programación
  - ✓ Presentan dificultades para verificación, validación, actualización y documentación de los modelos
  - ✓ No permiten modelar problemas complejos o de gran tamaño

Biblioteca de optimización en C, C++

Ventajas

- ✓ Tiempo de solución es crítico
- ✓ Permiten el uso de algoritmos de optimización específicos
- ✓ Posibilidad de implantación del modelo en un entorno software o hardware especial

Inconvenientes

- ✓ Mayor dificultad y consumo de recursos para el mantenimiento del modelo

Ventajas lenguajes algebraicos (i)

- Lenguajes de alto nivel para formulación compacta de modelos grandes y complejos
- Facilitan desarrollo de prototipos
- Mejorar productividad de modeladores
- Estructuran buenos hábitos de modelado
- Separan datos de estructura matemática de modelo
- Formulación independiente del tamaño
- Modelo independiente de optimizadores

Ventajas lenguajes algebraicos (ii)

- Facilitan reformulación continua
- Documentación simultánea al modelo
- Permiten construir grandes modelos “mantenibles” que se pueden adaptar rápidamente a situaciones nuevas
- Permiten implantación de algoritmos avanzados
- Implantación fácil de problemas NLP, MIP, MCP
- Portabilidad entre plataformas y sistemas operativos (MS Windows, Linux, Sun Solaris, HP UX, Digital True64Unix, IBM AIX, SGI IRIX, Mac OS X)

Desventajas lenguajes algebraicos

- No son adecuados para usos esporádicos con problemas de pequeño tamaño
- No son adecuados para resolución directa problemas de tamaño gigantesco (1.000.000 x 1.000.000)

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

Modelos matemáticos de optimización

Andrés Ramos
Andres.Ramos@iit.icai.upcomillas.es
Universidad Pontificia Comillas

Begoña Vitoriano
bvitoriano@mat.ucm.es
Universidad Complutense de Madrid