

# Formulando con modelos lineales enteros

Daniel Espinoza, Dpto. Ingeniería Industrial

Universidad de Chile

7 de marzo de 2011

# Contenidos

- 1 Forma de un problema Lineal Entero
- 2 Modelando con variables binarias
- 3 Tipos de Problemas

## Problema de optimización lineal entero (MILP)

$$\text{máx} \quad cx + hy \quad (1a)$$

$$\text{s.t.} \quad Ax + Gy \leq b \quad (1b)$$

$$x \in \mathbb{Z}^n, y \in \mathbb{R}^p \quad (1c)$$

- $x, y$  son las variables de decisión del problema.
- (1) se dice **mixto** por que tiene variables enteras y continuas.
- (1a) es la función objetivo (lineal en las variables).
- (1b) son las restricciones del dominio (lineales en las variables).
- (1c) el dominio general de las variables ( $\mathbb{R}^n$  y/o  $\mathbb{Z}^n$ ).
- Los  $S := \{x, y\}$  que satisfacen (1b) y (1c) es el **conjunto factible**.
- Si este conjunto es vacío, el problema se dice **infactible**.
- Una **instancia** de (1) se especifica por los datos  $(c, h, A, G, b)$ .
- Un  $(x_0, y_0) \in S$  es un **punto factible**.
- $(x_0, y_0) \in S$  es **óptimo** si  $cx_0 + hy_0 \geq cx + hy \quad \forall (x, y) \in S$ .
- (1) es equivalente a un problema de minimización, ¿por que?

# Algunas observaciones:

- Restricciones de igualdad, menor o igual, y mayor o igual, calzan la forma general de (1). ¿Como?
- Una instancia factible de (1) puede no tener solución óptima, ¿por que?
- En este caso la instancia se dice **no acotada**.
- **Resolver** una instancia de (1) es determinar si ésta es:
  - Infactible.
  - No acotada.
  - Óptima.
- En algunos casos (1c) se escribe como  $x \in \{0, 1\}^n, y \in \mathbb{R}^p$ , en estos casos (1) se dice **binario** (binary-MILP).

# Modelos lineales continuos:

- Modelaremos las siguientes situaciones usando tres variables positivas y continuas  $x, y, z$ .
  - “Por cada unidad de  $x$  deben haber al menos 5 unidades de  $y$ ”.
    - $x \leq \frac{1}{5}y$ .
  - “Por cada 7 unidades de  $x$  deben haber 9 unidades de  $y$ ”.
    - $\frac{1}{7}x = \frac{1}{9}y$ .
  - Un puerto puede cargar 11 unidades de  $x$ , o 45 de  $y$  o 30 de  $z$  por semana. ¿Qué combinaciones de  $x, y, z$  puede cargar en 10 semanas?
    - $\frac{1}{11}x + \frac{1}{45}y + \frac{1}{30}z \leq 10$ .
  - “Componente  $x$  debe ser al menos un 33 % de los componentes  $x, y, z$ ”
    - $x \geq 0,33(x + y + z) \iff 0,67x \geq 0,33(y + z)$
  - “Proceso I transforma una unidad de  $x$  en 24.5 unidades de  $y$  y 73.1 unidades de  $z$ ”
    - $y = 24,5x, \quad z = 73,1x$ .
  - “Si  $x, y, z$  tienen 30,90,5 % de  $L$ , ¿que mezclas de  $x, y, z$  tienen al menos 20 % de  $L$ ?”
    - $0,3x + 0,9y + 0,05z \geq 0,2(x + y + z)$ .

# Variables binarias para alternativas

- Situaciones donde uno u otro evento deben ocurrir.
- Para este tipo de dicotomías se usa:

$$x = \begin{cases} 1 & \text{evento ocurre} \\ 0 & \text{evento no ocurre} \end{cases}$$

## Problema de la Mochila (Knapsack)

Tenemos  $n$  posibles proyectos, cada uno requiere una inversión  $a_j$  y un retorno  $c_j$ , y tenemos una restricción presupuestaria de  $b$ . Los proyectos deben realizarse completamente para obtener el beneficio. ¿Qué proyectos deben llevarse a cabo?



$$\text{máx} \left\{ \sum_{i=1}^n c_i x_i : \sum_{i=1}^n x_i a_i \leq b, x \in \{0, 1\}^n \right\}$$

- ¿Qué tan distinta es la solución si  $x$  es continua?

# Variables binarias para alternativas

## Problema de asignación (Matching bi-partito)

Tenemos  $m$  trabajos y  $n$  personas a las cuales asignarlos ( $m \leq n$ ). Cada trabajo debe ser asignado a una persona, y cada persona puede trabajar a lo más en un trabajo. El costo de asignar trabajo  $j$  al trabajador  $i$  es  $c_{ij}$ . Queremos buscar la asignación de todos los trabajos a costo mínimo.

- Defina  $x_{ij} \in \{0, 1\}$  representando asociar trabajo  $j$  al trabajador  $i$ .
- ¿Que restricciones son necesarias?
- ¿Cuál es la cardinalidad de los puntos factibles?

## Matching perfecto

Tenemos  $2n$  trabajadores, y debemos formar  $n$  parejas de trabajo maximizando el beneficio global (de la empresa).

# Variables binarias para alternativas

## Set Covering, Set Packing, Set Partitioning

Consideremos un conjunto base  $\mathcal{M} = \{1, \dots, m\}$ , y poseemos una serie de sub-conjuntos  $\{\mathcal{M}_j\}_{j=1}^n$ ,  $\mathcal{M}_j \subseteq \mathcal{M}$ .

- Decimos que  $J \subseteq \{1, \dots, n\}$  **covering**  $\mathcal{M}$  si  $\bigcup_{i \in J} \mathcal{M}_i = \mathcal{M}$ .
- Decimos que  $J \subseteq \{1, \dots, n\}$  es un **packing** en  $\mathcal{M}$  si  $\mathcal{M}_i \cap \mathcal{M}_j = \emptyset \quad \forall i, j \in J$ .
- Decimos que  $J \subseteq \{1, \dots, n\}$  es una **partición** en  $\mathcal{M}$  si es un covering y un packing a la vez.

¿Cómo describimos todos los  $J$  que son un *covering*, *packing*, *partition* en  $\mathcal{M}$ ?

### Ejemplo:

Consideremos el problema de colocar estaciones de bomberos en una ciudad, y queremos asegurar un tiempo de respuesta de no más de 10 minutos? y si queremos  $K$  cobertura?

# Modelando otras relaciones

- Ya modelamos como a lo más un evento puede ocurrir:
  - $\sum_{i=1}^n x_i \leq 1.$
- Si dos eventos deben ocurrir en forma simultanea o no ocurrir:
  - $x_i - x_j = 0.$
- Si un evento es necesario para que otro ocurra:
  - $x_i \leq x_j.$
- $x$  ocurre si y sólo si  $y$  y  $z$  ocurren:
  - $x \leq y, \quad x \leq z, \quad x + 1 \geq y + z.$
- $x$  ocurre si y sólo si  $y$  o  $z$  ocurren:
  - $x \geq y, x \geq z, x \leq y + z.$
- Consideremos una actividad que puede desarrollarse dentro de niveles  $0 \leq y \leq u$ , pero sólo si el evento  $x$  ocurre:
  - $0 \leq y, y \leq ux.$

# Modelando otras relaciones

## Problema de ubicación de instalaciones

Consideremos  $I = \{1, \dots, m\}$  un conjunto de clientes, y un conjunto de posibles locación de instalaciones  $N = \{1, \dots, n\}$ . Abrir instalación  $j \in N$  cuesta  $c_j$ . Atender al cliente  $i \in I$  desde  $j \in N$  cuesta  $h_{ij}$ .

- ¿Cómo modelamos el problema?
- ¿Que pasa si hay capacidades por instalación  $\{b_j : j \in N\}$  y demanda  $\{d_i : i \in I\}$ ?
- ¿Y si además hay costo variable  $\{v_{ij} : i \in I, j \in N\}$  dependiendo de la demanda?

# Modelando otras relaciones

## Flujo en redes con costo fijo

Consideremos una **red** o **grafo**  $G = (V, E)$ , donde  $V = \{1, \dots, n\}$  se llaman **nodos** y  $E \subseteq V \times V$  se llaman **arcos** o **aristas**.

Si  $e = (u, v) \in E$  entonces existe una ruta directa para ir de  $u$  a  $v$  (pero no necesariamente de  $v$  a  $u$ ).

Asociado a cada nodo  $v \in V$  existe una demanda  $d_v$ . Un nodo  $v$  es una fuente, punto de tránsito o punto de demanda dependiendo si  $d_v$  es negativo, cero o positivo respectivamente. Asumimos que la demanda neta es cero ( $\sum_{v \in V} d_v = 0$ ).

Cada arco  $e \in E$  tiene una capacidad  $u_e \geq 0$  y un costo de transporte unitario  $h_e$ .

- ¿Cómo definiría un **flujo** factible para la red?
- ¿Cómo modelaría la existencia de costos fijos  $c$  por uso de rutas (arcos) y/o nodos?

# Modelando otras relaciones

## Problema del Vendedor Viajero

Consideramos un **grafo**  $G = (V, E)$ . Los nodos representan ciudades y los arcos  $e = (u, v) \in E$  la posibilidad de viaje directo entre las ciudades  $u$  y  $v$ . Cada arco  $e \in E$  tiene asociado un costo  $c_e$ .

El objetivo es encontrar un **tour** que, partiendo de la ciudad 1, visita todas las ciudades exactamente una vez volviendo a la ciudad de partida, a costo mínimo.

- Usando variables por arco, formule el problema.
- ¿Cuántas restricciones necesitamos?
- ¿Cuál es el número mínimo necesario?
- ¿Hay formulaciones más compactas?

# Funciones no-lineales y restricciones disyuntivas

- ¿Qué pasa si la función objetivo no es lineal?
  - Consideraremos funciones objetivos del tipo  $f(y_1, \dots, y_N) = \sum_{i \in N} f_i(y_i)$  no lineal.
  - $f$  se dice **separable**.
  - Cada  $f_i$  es no lineal.
  - Basta considerar  $f_i$  lineal por trazo?
  - ¿por que?
  - ¿Qué pasa si  $f$  no es separable?
- ¿Qué sucede si en vez de requerir que **todas** las restricciones se cumplan, nos basta que se cumplan  $k$  de ellas?
  - ¿Por qué podría ser esto interesante?
  - Este tipo de relaciones se llaman **disyuntivas**.

# Funciones lineales por trazo

- Asumamos que  $f$  es lineal por trazo y continua:
  - $f$  puede ser descrita por los puntos  $\{a_i, f(a_i)\}_{i=1}^n$  donde  $a_i \leq a_{i+1}$ .
  - Note que todo  $a_1 \leq y \leq a_r$  puede escribirse como
 
$$y = \sum_{i=1}^n \lambda_i a_i \quad \text{con} \quad \sum_{i=1}^n \lambda_i = 1 \quad y \quad \lambda \geq 0.$$
  - Dado  $y$ , la representación en términos de  $\lambda$  no es única.
  - Si embargo, si  $a_i \leq y \leq a_{i+1}$ , podemos escoger  $\lambda$  tal que
 
$$\lambda_j = 0 \quad \forall j \neq \{i, i+1\}. \text{ Así, } y = \lambda_i a_i + \lambda_{i+1} a_{i+1} \text{ y } \\ f(y) = \lambda_i f(a_i) + \lambda_{i+1} f(a_{i+1})$$
  - Bajo este supuesto, se tendrá  $f(y) = \sum_{i=1}^n \lambda_i f(a_i)$ .
  - Ahora, definiendo el evento “ $y$  esta en el intervalo  $[a_i, a_{i+1}]$ ”, modele las condiciones anteriores utilizando variables binarias.
  - Si minimizamos  $f$  lineal por trozo convexa, las variables binarias anteriores no son necesarias.
  - ¿Qué sucede si  $f$  es discontinua?
  - ¿Hay otras formulaciones? ¿Son mejores? ¿En qué sentido?
  - ¿Podemos, en general, usar menos que  $n - 1$  variables binarias?

# Restricciones Disyuntivas

- Consideremos la siguiente situación: queremos definir  $y = \min\{u_1, u_2\}$ .
  - Podemos modelar esto como  $y \leq u_1, y \leq u_2$  y la restricción disyuntiva  $y \geq u_1$  o  $y \geq u_2$ .
- En general queremos describir un punto satisfaciendo al menos  $k$  de  $m$  restricciones, en el ejemplo  $m = 2, k = 1$ .
  - Supongamos  $P^i := \{y \in \mathbb{R}_+^p : A^i y \leq b^i, y \leq d\}$  para  $i = 1, \dots, m$ .
  - Note que  $P^i \subseteq \{y \in \mathbb{R}^p : 0 \leq y \leq d\} = Y$ .
  - Implica que existe  $w^i$  tal que  $A^i y \leq b^i + w^i \quad \forall y \in Y$ .
  - Usando el evento " $y \in P^i$ ", modele la restricción disyuntiva general.
- Para el caso  $k = 1$  existe una formulación alternativa:
  - Consideramos  $\sum_{i=1}^m y^i = y, \quad \sum_{i=1}^m x_i = 1$ .
  - ¿Cuál formulación es mejor?
- ¿Qué pasa si queremos modelar que a lo más  $k$  de  $m$  conjuntos de restricciones se satisfacen?

# Restricciones Disyuntivas

## Problema de Secuenciamiento

Considere que tenemos  $n$  trabajos y  $m$  máquinas. Cada trabajo debe pasar por cada máquina. Para cada trabajo, el orden en que pasa por cada máquina es pre-establecido, es decir, el trabajo  $j$  debe partir por la máquina  $j(1)$  y terminar en la máquina  $j(m)$ . Cada trabajo  $j$  demora un tiempo  $p_{jk}$  en la máquina  $k$ . Cada máquina puede procesar un solo trabajo, y una vez comenzado un trabajo, debe finalizarlo. El objetivo es minimizar la suma de los tiempos de finalización de los trabajos.

- Sea  $t_{jk}$  el tiempo de término del trabajo  $j$  en la máquina  $k$ .
  - $t_{j,j(r+1)} \geq t_{j,j(r)} + p_{j,j(r+1)} \quad \forall j, r = 1, \dots, m-1$ .
  - Dada una máquina  $k$  y dos trabajos  $i, j$ , siempre es cierto que “ $i$  se procesa antes que  $j$ ” o “ $j$  se procesa antes que  $i$ ”.
  - Definimos  $x_{ijk} = 1$  si lo primero es cierto, y cero si no.
  - Necesitamos  $m \binom{n}{2}$  variables binarias  $\mathcal{O}(m \binom{n}{2})$  restricciones.
  - Modelo alternativo para el TSP. ¿Cómo?
  - Difícil de resolver para problemas de interés práctico.

# Dificultad de los Problemas

- La dificultad de un problema está relacionada con la estructura del problema y con el tamaño de la instancia.
- Se suele estudiar el número de operaciones aritméticas elementales (comparaciones, sumas, multiplicaciones) en función del tamaño de la instancia que un algoritmo realiza para resolver un problema.

## Algoritmo Polinomial

El número de operaciones está acotado por un polinomio en el tamaño del problema. Estos algoritmos se consideran eficientes.

## Algoritmo Exponencial

El número de operaciones no está acotado por un polinomio. Estos algoritmos se consideran ineficientes.

Son todos los problemas iguales?

# Dificultad de los Problemas

## P

Problemas en los cuales existe un algoritmo polinomial que los resuelve (problemas fáciles).

## NP (non-deterministic polinomial)

Problemas en los cuales existe un algoritmo no determinístico que lo resuelve en tiempo polinomial. Entre los problemas NP existe una subclase, los problemas NP-completos, que son los difíciles aún no resueltos eficientemente.