

Inteligencia Artificial

1 Semestre 2018

Elizabeth Montero Ureta

Departamento de Informática
Universidad Técnica Federico Santa María

1er Semestre 2018

- Optimización: Determinación de una alternativa de decisión con la propiedad de ser mejor que cualquier otra en algún sentido a precisar
 - ▶ Función objetivo: Medida cuantitativa del funcionamiento del sistema que se desea optimizar
 - ▶ Variables: Decisiones que se pueden tomar para afectar el valor de la función objetivo
 - ▶ Restricciones: Relaciones (ecuaciones e inecuaciones) que las variables están obligadas a cumplir
 - ▶ Constantes/Parámetros: Atributos del problema conocidos a priori y fijos que permiten simplificar la formulación del modelo
- Resolver: Encontrar valor de las variables que optimiza la función objetivo y satisface todas las restricciones. (Detectar que el problema no tiene solución)

Importancia del Modelado

- Entender el problema
- Servir como mecanismo de comunicación
- Identificar semejanzas con problemas clásicos
 - ▶ Mecanismos de resolución

Clasificación de modelos de optimización

- Programación lineal (continua)
- Programación lineal entera mixta
- Programación no lineal

- Las necesidades mínimas en la alimentación de una ternera son de 700 [g] de proteínas, 28 [g] de calcio y 150 [mg] de vitaminas. Los alimentos disponibles son pienso¹ y forraje con un costo de 0.30 y 0.35 euros/kg respectivamente. La composición nutritiva por kg:

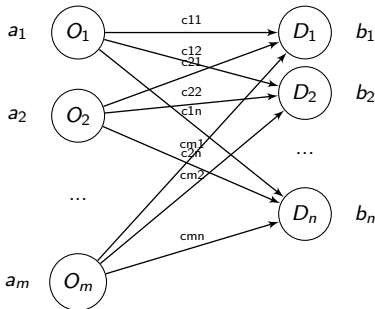
	Proteínas [g]	Calcio [g]	Vitaminas [mg]
Pienso	30	2	10
Forraje	45	1	5

- Se trata de determinar la cantidad diaria óptima de cada alimento para minimizar el costo total de alimentación.

¹Alimento seco que se le da al ganado

Problemas de Transporte

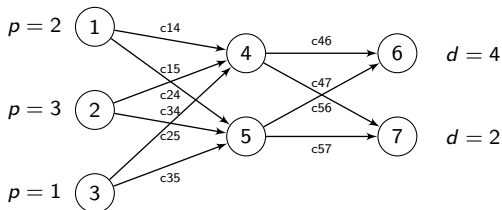
- Minimizar el costo total de transporte de un producto desde ciertos orígenes a ciertos destinos satisfaciendo la demanda de cada destino sin superar la oferta disponible en cada origen.
- Se supone todos los orígenes conectados con todos los destinos:



- Se busca satisfacer la demanda sin superar la oferta a mínimo costo

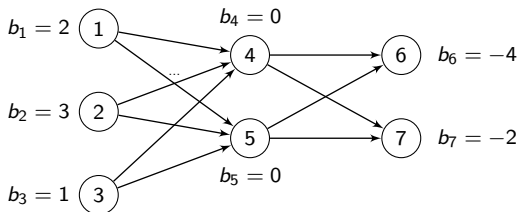
Problemas de Transbordo

- Llevar un producto desde orígenes a destinos con puntos intermedios en una red de n nodos con mínimo costo.



Problemas de Transbordo

- Llevar un producto desde orígenes a destinos con puntos intermedios en una red de n nodos con mínimo costo.



- Considerando:
 - $b_i > 0 \rightarrow$ nodo origen (generador)
 - $b_i < 0 \rightarrow$ nodo destino (consumidor)
 - $b_i = 0 \rightarrow$ nodo transbordo (no genera ni consume)

Problemas de Transbordo

- Variables:

x_{ij} : cantidad transportada desde el nodo i al nodo j

- Constantes:

n : total de nodos

c_{ij} : costo (por kilogramo) de transportar desde el nodo i al nodo j

b_i : flujo en el nodo i

$$\text{Mín} \sum_i^n \sum_j^n c_{ij} * x_{ij}$$

$$\sum_j^n x_{ij} - \sum_j^n x_{ji} = b_i \quad \forall i = 1, \dots, n$$

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

Modelado con variables enteras y binarias

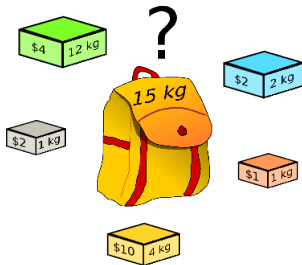
El uso de variables enteras y binarias aumenta considerablemente las posibilidades de modelado:

- Modelado de cantidades discretas
- Modelado de decisiones que implican un costo fijo o de arranque:
 - ▶ Adquirir o no un activo (un edificio, una máquina, etc.)
 - ▶ Poner en marcha un proceso o no.
- Modelado de decisiones que posibilitan la toma de otras decisiones:
 - ▶ La compra de un determinado aparato (variable binaria) permite después tomar decisiones relativas a su operación.
- Modelado de restricciones no lineales y no convexas.
- Modelado de implicaciones y de condiciones lógicas.

Problema de la Mochila (Knapsack)

Modelos tipo Problema de la Mochila

Problema de la mochila (Knapsack problem) modela una situación análoga al llenado de una mochila, incapaz de soportar más de un peso determinado, con todo o parte de un conjunto de objetos, cada uno con un peso y valor específicos. Los objetos colocados en la mochila deben maximizar el valor total sin exceder el peso máximo.



Modelos de Presupuesto [Mochila Multidimensional]

- El límite total de recursos disponibles consumidos por los proyectos seleccionados e inversiones en cada período de tiempo no deben exceder los recursos disponibles.

Restricciones en un Modelo de Presupuesto:

- Actividades mutuamente excluyentes
La actividad 4 no se puede realizar si se realiza la actividad 5 y viceversa
- Dependencia entre actividades
La actividad 11 requiere que la actividad 2 se realice

- Espacio de búsqueda
- Brute-force search, generate and test
- Explosión combinatoria

Modelos de Set Covering, Packing y Partitioning

Modelos de Set Covering, Packing y Partitioning

Trabajan con un conjunto de elementos, la idea es construir sub-conjuntos de elementos considerando diversos puntos de vista respecto a las restricciones:

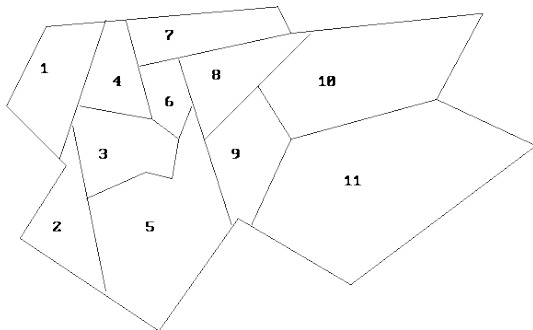
Set Covering Requieren que cada elemento pertenezca a **al menos** un sub-conjunto.

Set Packing Requieren que cada elemento aparezca en **a lo más** un sub-conjunto.

Set Partitioning Requieren que cada elemento aparezca en **exactamente** un sub-conjunto.

Set Covering

Instalar la menor cantidad de Estaciones de Bomberos de manera que puedan satisfacer las demandas de las 11 comunas de una región. Se considera que una Estación de Bomberos es capaz de satisfacer las demandas de la comuna en la que se encuentra y de las inmediatamente adyacentes a dicha comuna.



Considere un conjunto S de personas:

$$S = \{1, 2, 3, 4, 5\}$$

Suponga que desea organizar dichas personas en varios equipos conocidos:

- El conjunto de equipos posibles es s
- Por ejemplo: $s = \{\{1, 2\}, \{4, 5\}, \{1, 3, 5\}, \{2, 4, 5\}, \{1\}, \{3\}\}$
- Cada equipo tiene un costo c_j
- Se desea elegir los equipos que permiten tener ocupadas a **todas** las personas con un costo mínimo
 - ▶ No importa que la misma persona este en más de un equipo

Set Partitioning

Considere un conjunto S de personas:

$$S = \{1, 2, 3, 4, 5\}$$

Suponga que desea organizar dichas personas en varios equipos conocidos:

- El conjunto de equipos posibles es s
- Por ejemplo: $s = \{\{1, 2\}, \{4, 5\}, \{1, 3, 5\}, \{2, 4, 5\}, \{1\}, \{3\}\}$
- Cada equipo tiene un costo c_j
- Se desea elegir los equipos que permiten que cada persona esté **exactamente en un equipo** con un costo mínimo

Considere un conjunto S de personas:

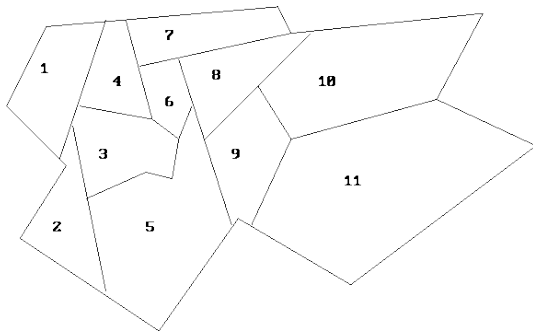
$$S = \{1, 2, 3, 4, 5\}$$

Suponga que desea organizar dichas personas en varios equipos conocidos:

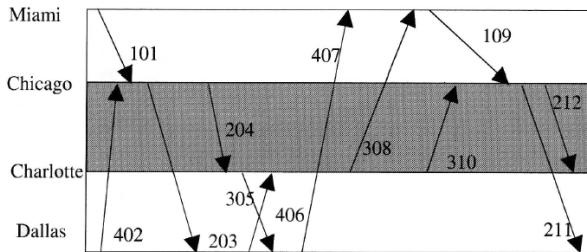
- El conjunto de equipos posibles es s
- Por ejemplo: $s = \{\{1, 2\}, \{4, 5\}, \{1, 3, 5\}, \{2, 4, 5\}, \{1\}, \{3\}\}$
- Cada equipo tiene un beneficio b_j
- Se desea elegir los equipos que proporcionen el beneficio máximo **sin que se traslapen**.
 - ▶ Que las personas no se repitan entre equipos diferentes, no importa que una persona no esté en ningún equipo

Set Covering

Instalar dos Estaciones de Bomberos de manera que puedan satisfacer las demandas de las comunas más importantes de una ciudad. La importancia de la comuna viene dada por la cantidad de ciudadanos que viven en dicha comuna. Se considera que una Estación de Bomberos es capaz de satisfacer las demandas de la comuna en la que se encuentra y de las inmediatamente adyacentes a dicha comuna.



AA Crew Scheduling



Generación de Columnas en AA Crew Scheduling

Paso 1:

Generación de una secuencia de columnas

j	Secuencia de Vuelos	Costo
1	101 - 203 - 406 - 308	2900
2	101 - 203 - 407	2700
3	101 - 204 - 305 - 407	2600
4	101 - 204 - 308	3000
5	203 - 406 - 310	2600
6	203 - 407 - 109	3150
7	204 - 305 - 407 - 109	2550
8	204 - 308 - 109	2500
9	305 - 407 - 109 - 212	2600
10	308 - 109 - 212	2300
11	310 - 212	2000
12	402 - 203	2100
13	402 - 204 - 305	2400
14	402 - 204 - 310 - 211	2550
15	406 - 308 - 109 - 211	2750
16	406 - 310 - 211	2600
17	407 - 109 - 211	2550

- La generación de columnas se utiliza como una estrategia de dos pasos para enfrentar la resolución de problemas combinatorios altamente complejos.
- Consiste en la generación de una secuencia de columnas donde cada columna representa una alternativa factible (posible parte de una solución)
- La selección del conjunto óptimo de alternativas se resuelve mediante un set Partitioning (o Covering)
- Ventaja: Flexibilidad
- Desventaja: Dificultad para enumerar todas las columnas.

Generación de Columnas en AA Crew Scheduling

Paso 2:

Resolver el problema como un Set [Covering/ Packing/ Partitioning]

Variables:

$$x_j = \begin{cases} 1 & \text{Si se elije la columna } j \\ 0 & \text{si no} \end{cases}$$

Constantes:

$$a_{ij} = \begin{cases} 1 & \text{Si el vuelo } i \text{ está en la columna } j \\ 0 & \text{si no} \end{cases}$$

C_j = Costo de la columna j

Objetivo:

$$\text{Mín } \sum_{j=1}^{17} C_j \cdot x_j$$

Restricciones:

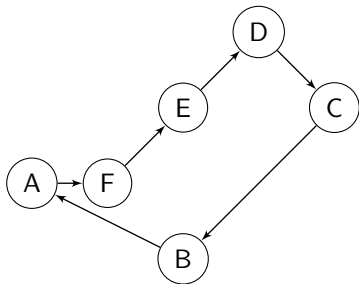
$$\sum_{j=1}^{17} a_{ij} \cdot x_j = 1 \quad \forall i$$

Solución $x_1 = x_9 = x_{14} = 1$ a un costo total de 8050

Problema del vendedor viajero

Problema del vendedor viajero

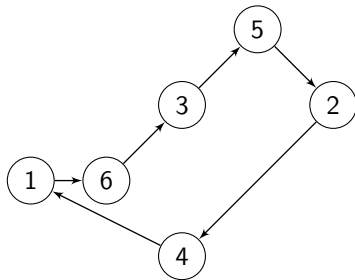
El problema del vendedor viajero (Traveling Salesman Problem (TSP)) consiste en encontrar un circuito de costo mínimo que pase una sola vez por cada ciudad que debe visitar el vendedor y que le permita volver a su ciudad de origen al final del día.



- Problema simétrico versus asimétrico

Problema del vendedor viajero

El problema del vendedor viajero (Traveling Salesman Problem (TSP)) consiste en encontrar un circuito de costo mínimo que pase una sola vez por cada ciudad que debe visitar el vendedor y que le permita volver a su ciudad de origen al final del día.



- Problema simétrico versus asimétrico

Problema del vendedor viajero

Problema de Optimización: TSP

Datos: Dado un grafo G de orden n , completo y valorizado con valores positivos

Objetivo: Determinar el largo mínimo de un ciclo hamiltoniano en G

Problema de reconocimiento/decisión: RTSP

Datos: Dado un entero k , dado un grafo G de orden n , completo y valorizado con valores positivos

Pregunta: Existe en el grafo un ciclo hamiltoniano de largo inferior o igual a k ?

RTSP permite resolución en tiempo polinomial ssi TSP también lo permite

Problemas P versus NP

Problema P

Un problema se dice polinomial si existe un algoritmo de complejidad polinomial que permite responder la pregunta del problema cualquiera sea el dato de éste. La clase P es el conjunto de todos los problemas de reconocimiento polinomiales.

Problemas NP (No-determinístico polinomiales)

Un problema de reconocimiento está en la clase NP si, para toda instancia de ese problema, en un tiempo polinomial con respecto al tamaño de la instancia, se puede verificar una solución propuesta o adivinada (“sí”)

$P \subseteq NP$

No existe certeza si las clases P y NP coinciden o si la inclusión de la clase P en NP es estricta.

Problemas NP-completos

Transformación polinomial

Dados dos problemas de reconocimiento: D_1 y D_2 , $D_1 \prec D_2$ ssi:

- Existe una aplicación f que transforme cualquier instancia I de D_1 en una instancia $f(I)$ de D_2 , y un algoritmo polinomial, con respecto al tamaño de I para calcular $f(I)$.
- Hay una equivalencia entre los dos enunciados " D_1 acepta la respuesta "sí" para la instancia I " y " D_2 acepta la respuesta "sí" para la instancia $f(I)$ ".

Si $D_1 \prec D_2$ y existe un algoritmo polinomial para resolver D_2 , luego existe un algoritmo polinomial para resolver D_1 .

Además se cumple que si $D_1 \prec D_2$ y $D_2 \prec D_3$, entonces $D_1 \prec D_3$.

NP-completitud

Un problema Q se dice NP-completo si pertenece a la clase NP y si, para todo problema Q' de la clase NP, se tiene que $Q' \prec Q$.

Por qué demostrar la NP-completitud?

- Evitar perder tiempo buscando un algoritmo polinomial que resuelva el problema
- Justificar el uso de heurísticas para el problema que entreguen resultados cercanos al óptimo.

Aquí el interés no se centra sólo en los problemas de reconocimiento.

NP-difícil

Sea O un problema de optimización. Si el problema de reconocimiento asociado a O es NP-completo, entonces O es NP-difícil.

El RTSP es NP-completo y el TSP es NP-difícil.

Modelos de Asignación, redes y ruteo de vehículos

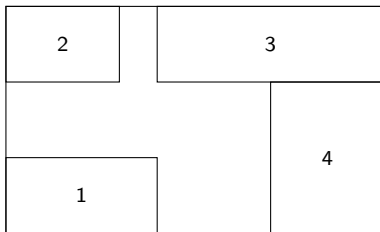
Modelos de Problema de Asignación

- Encontrar la mejor asignación (máquina-trabajo, personal-cliente) para minimizar costos

Modelos de Problema de Asignación Cuadrática

Ejemplo: Mall Layout

Se tienen 4 posibles ubicaciones para departamentos en un shopping mall. Se conocen las distancias (en metros) entre las ubicaciones. Se conoce además el número de clientes a la semana que desearían visitar los diferentes pares de departamentos. Por ejemplo, se proyecta que 500 clientes a la semana visitarían la tienda de ropa (Tienda 1) y computación (Tienda 2). El objetivo del problema es determinar la ubicación de las tiendas minimizando la molestia de los clientes.



- Planificación óptima para una colección dada de trabajos. Cada uno de dichos trabajos requiere una secuencia de procesadores, los cuales pueden realizar sólo un trabajo a la vez.
- Suponga tres trabajos a realizar: W_1 , W_2 y W_3 .

W_1	
p_2	3
p_3	3
p_4	3
...	...
p_8	2

W_2	
p_3	7
p_1	2
...	...
p_9	6

W_3	
p_7	5
p_6	9
p_3	2
p_5	1
...	...
p_{10}	5

Problema de ruteo de vehículos

El problema de ruteo de vehículos (Vehicle routing problem (VRP)) busca satisfacer la demanda de un conjunto de clientes utilizando una flota de vehículos. Los bienes son despachados desde un almacén central a todos los consumidores. El objetivo consiste en construir las rutas que deben realizar los vehículos de la flota de modo de minimizar los costos de entrega (tiempo, bencina).

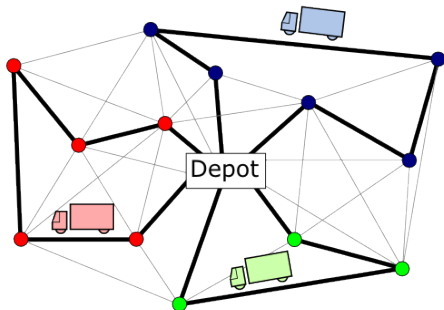


Figura: <http://www.liacs.nl/~ftakes/vrp/>

Modelado de restricciones especiales

Disyunción

Restricción:

De dos restricciones al menos una debe darse. Debe cumplirse una de las restricciones, no necesariamente las dos.

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

Variables:

$$\delta = \begin{cases} 1 & \text{obliga a } g(x) \leq 0 \text{ y relaja la otra restricción} \\ 0 & \text{obliga a } f(x) \leq 0 \text{ y relaja la otra restricción} \end{cases}$$

Restricciones:

$$\begin{aligned} f(x) &\leq M_1 \cdot \delta \\ g(x) &\leq M_2 \cdot (1 - \delta) \end{aligned}$$

Cumplir k de N ecuaciones

Restricción:

De N ecuaciones se deben cumplir al menos k , siendo $k < N$

$$\{f_1(x) \leq 0\}, \dots, \{f_N(x) \leq 0\}$$

Variables:

$$y_i = \begin{cases} 1 & \text{si la restricción } i \text{ se cumple} \\ 0 & \text{si no} \end{cases}$$

Restricciones:

$$\begin{aligned} f_i(x) &\leq M_i \cdot (1 - y_i) \\ \sum_{i=1}^N y_i &\geq k \end{aligned}$$

Seleccionar el valor de una función entre N valores posibles

Restricción: Seleccionar entre N valores: Una ecuación con múltiples valores posibles.

$$f(x) = \begin{cases} v_1 \\ \dots \\ v_N \end{cases}$$

Variables:

$$y_i = \begin{cases} 1 & \text{si la ecuación toma el valor } i\text{-ésimo} \\ 0 & \text{si no} \end{cases}$$

Restricciones:

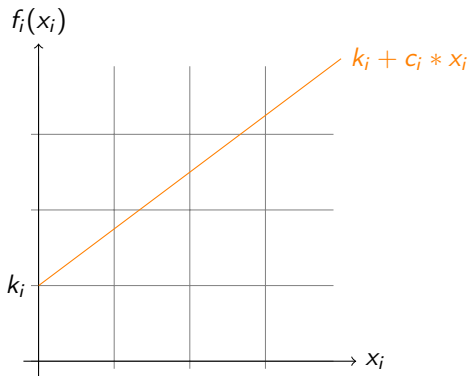
$$f(x) = \sum_{i=1}^N v_i \cdot y_i$$

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

Modelado de costos fijos (1/2)

- Costo con un término fijo si la variable toma un valor estrictamente positivo

$$f_i(x_i) = \begin{cases} 0 & x_i = 0 \\ k_i + c_i * x_i & x_i > 0 \end{cases}$$



- Variable auxiliar binaria:

$$y_i = \begin{cases} 1 & x_i > 0 \\ 0 & x_i = 0 \end{cases}$$

$$\begin{array}{ll} \text{Mín} & k_i * y_i + c_i * x_i \\ x_i \leq M * y_i & (M \text{ valor grande}) \end{array}$$

Modelado de funciones minimax (1/2)

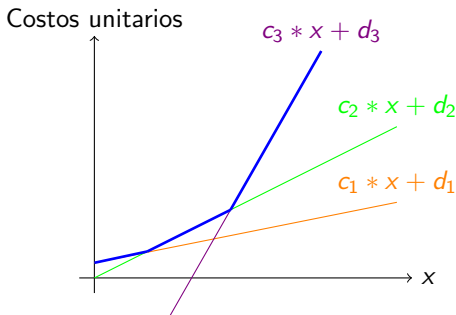
- Modelado de la minimización del máximo costo:

$$\text{Mín } f(x), A * x = b, x \geq 0$$

donde

$$f(x) = \text{Máx} \{ \{c_1 * x + d_1\}, \{c_2 * x + d_2\}, \dots, \{c_p * x + d_p\} \}$$

$$x, c_i \in \mathbb{R}^n, d_i \in \mathbb{R}$$



- Gráficamente: En una dimensión

Mín z

$$z \geq c_1 * x + d_1$$

$$z \geq c_2 * x + d_2$$

...

$$z \geq c_p * x + d_p$$

$$A * x = b$$

$$x \geq 0$$

Modelado de implicaciones lógicas

Caso 1.a: De variables binarias a restricciones \leq

Restricción

IF $\delta = 1$ THEN $\sum_i a_i \cdot x_i \leq b$

Formulación

$$\sum_i a_i \cdot x_i \leq b + (1 - \delta) \cdot M$$

Consecuencia

Como $A \rightarrow B$ equivale a $\neg B \rightarrow \neg A$

IF $\sum_i a_i \cdot x_i > b$ THEN $\delta = 0$

Caso 1.b: De variables binarias a restricciones \geq

Restricción

IF $\delta = 1$ THEN $\sum_i a_i \cdot x_i \geq b$

Formulación

$$\sum_i a_i \cdot x_i \geq b + (1 - \delta) \cdot m$$

Consecuencia

Como $A \rightarrow B$ equivale a $\neg B \rightarrow \neg A$

IF $\sum_i a_i \cdot x_i < b$ THEN $\delta = 0$

Caso 2.a: De restricciones \geq a variables binarias

Restricción

IF $\sum_i a_i \cdot x_i \geq b$ THEN $\delta = 1$

Equivalencia

IF $\sum_i a_i \cdot x_i \geq b$ THEN $\delta = 1$ equivale a IF $\delta = 0$ THEN $\sum_i a_i \cdot x_i < b$
IF $\delta = 0$ THEN $\sum_i a_i \cdot x_i \leq b - \epsilon$

Formulación

$\sum_i a_i \cdot x_i \leq b - \epsilon + \delta \cdot M'$

Consecuencia

Como $A \rightarrow B$ equivale a $\neg B \rightarrow \neg A$
IF $\delta = 0$ THEN $\sum_i a_i \cdot x_i < b$

Caso 2.b: De restricciones \leq a variables binarias

Restricción

IF $\sum_i a_i \cdot x_i \leq b$ THEN $\delta = 1$

Equivalencia

IF $\sum_i a_i \cdot x_i \leq b$ THEN $\delta = 1$ equivale a IF $\delta = 0$ THEN $\sum_i a_i \cdot x_i > b$
IF $\delta = 0$ THEN $\sum_i a_i \cdot x_i \geq b + \epsilon$

Formulación

$\sum_i a_i \cdot x_i \geq b + \epsilon + \delta \cdot m'$

Consecuencia

Como $A \rightarrow B$ equivale a $\neg B \rightarrow \neg A$
IF $\delta = 0$ THEN $\sum_i a_i \cdot x_i > b$

Caso 3: De variables binarias a restricciones =

Restricción

IF $\delta = 1$ THEN $\sum_i a_i \cdot x_i = b$

Equivalencia

IF $\delta = 1$ THEN $\sum_i a_i \cdot x_i = b$ equivale a:

IF $\delta = 1$ THEN $\sum_i a_i \cdot x_i \leq b$ AND $\sum_i a_i \cdot x_i \geq b$

Formulación

$$\sum_i a_i \cdot x_i \geq b + (1 - \delta) \cdot m$$

$$\sum_i a_i \cdot x_i \leq b + (1 - \delta) \cdot M$$

Caso 4: De restricciones = a variables binarias

Restricción

IF $\sum_i a_i \cdot x_i = b$ THEN $\delta = 1$

Equivalencia

SI $\sum_i a_i \cdot x_i = b$ THEN $\delta = 1$ equivale a:

SI $(\sum_i a_i \cdot x_i \leq b \text{ AND } \sum_i a_i \cdot x_i \geq b)$ THEN $\delta = 1$

SI $\delta = 0$ THEN $(\sum_i a_i \cdot x_i < b \text{ OR } \sum_i a_i \cdot x_i > b)$

Formulación

$$\sum_i a_i \cdot x_i \geq b + \epsilon + \delta_2 \cdot m'$$

$$\sum_i a_i \cdot x_i \leq b - \epsilon + \delta_1 \cdot M'$$

$$\delta_1 + \delta_2 - 1 \leq \delta$$

Caso 5: De restricciones a restricciones (1)

Hint!: Usar que $A \Rightarrow B \Leftrightarrow B \vee \neg A$

Caso 5.1

$$\sum_i a_i \cdot x_i > b \Rightarrow \sum_i c_i \cdot x_i \leq d$$
$$\{\sum_i c_i \cdot x_i \leq d\} \vee \{\sum_i a_i \cdot x_i \leq b\}$$

Caso 5.2

$$\sum_i a_i \cdot x_i \geq b \Rightarrow \sum_i c_i \cdot x_i \leq d$$
$$\{\sum_i c_i \cdot x_i \leq d\} \vee \{\sum_i a_i \cdot x_i \leq b - \epsilon\}$$

Caso 5.3

$$\sum_i a_i \cdot x_i = b \Rightarrow \sum_i c_i \cdot x_i \geq d$$
$$\{\sum_i c_i \cdot x_i \geq d\} \vee \{\sum_i a_i \cdot x_i \leq b - \epsilon\} \vee \{\sum_i a_i \cdot x_i \geq b + \epsilon\}$$

Caso 5: De restricciones a restricciones (2)

Caso 5.4

$$\sum_i a_i \cdot x_i \geq b \Rightarrow \sum_i c_i \cdot x_i = d$$
$$\{\{\sum_i c_i \cdot x_i \leq d\} \text{ AND } \{\sum_i c_i \cdot x_i \geq d\}\} \text{ OR } \{\sum_i a_i \cdot x_i \leq b - \epsilon\}$$

Formulación

$$\sum_i c_i \cdot x_i \leq d + \delta \cdot M$$
$$\sum_i c_i \cdot x_i \geq d + \delta \cdot m$$
$$\sum_i a_i \cdot x_i \leq b - \epsilon + (1 - \delta) \cdot M'$$

Modelado de dobles implicaciones

Variables binarias y restricciones \leq

$$\delta = 1 \Leftrightarrow \sum_i a_i \cdot x_i \leq b$$

$$\text{IF } \delta = 1 \text{ THEN } \sum_i a_i \cdot x_i \leq b$$

$$\sum_i a_i \cdot x_i \leq b + (1 - \delta) \cdot M$$

$$\text{IF } \sum_i a_i \cdot x_i \leq b \text{ THEN } \delta = 1$$

$$\sum_i a_i \cdot x_i \geq b + \epsilon + \delta \cdot m'$$

Variables binarias y restricciones \geq

$$\delta = 1 \Leftrightarrow \sum_i a_i \cdot x_i \geq b$$

$$\text{IF } \delta = 1 \text{ THEN } \sum_i a_i \cdot x_i \geq b$$

$$\sum_i a_i \cdot x_i \geq b + (1 - \delta) \cdot m$$

$$\text{IF } \sum_i a_i \cdot x_i \geq b \text{ THEN } \delta = 1$$

$$\sum_i a_i \cdot x_i \leq b - \epsilon + \delta \cdot M'$$

Restricciones & restricciones

$$\sum_i a_i \cdot x_i \geq b \Leftrightarrow \sum_i c_i \cdot x_i \leq d$$

$$\text{IF } \sum_i a_i \cdot x_i \geq b \text{ THEN } \sum_i c_i \cdot x_i \leq d$$

$$\text{IF } \sum_i c_i \cdot x_i \leq d \text{ THEN } \sum_i a_i \cdot x_i \geq b$$

$$\begin{aligned} \sum_i c_i \cdot x_i &\leq d + \delta \cdot K \\ \sum_i a_i \cdot x_i &\leq b - \epsilon + (1 - \delta) \cdot M' \end{aligned}$$

$$\begin{aligned} \sum_i a_i \cdot x_i &\geq b + \delta \cdot m \\ \sum_i c_i \cdot x_i &\geq d + \epsilon + (1 - \delta) \cdot k' \end{aligned}$$