Tema 2: Modelos lineales de optimización con variables enteras.

Objetivos del tema:

- ❑ Introducir la programación lineal entera y los dominios de aplicación.
- ❑ Aprender a formular el modelo de un problema de programación lineal entera.
- ❑ Modelar relaciones lógicas entre restricciones con variables binarias
- ❑ Expresar un problema entero general como un problema binario
- ❑ Modelar matemáticamente y resolver en OPL varios problemas típicos de programación lineal entera
- ❑ Expresar algunos comportamientos no lineales como problemas enteros

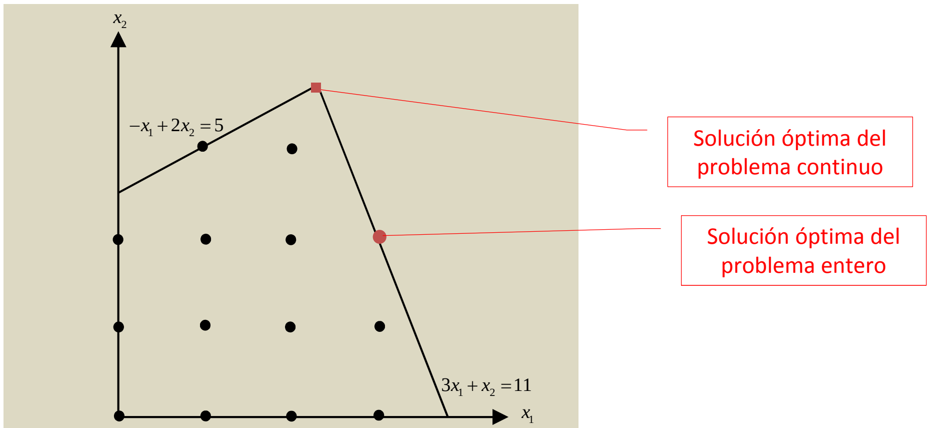
Introducción

En muchos problemas de programación lineal sólo tienen sentido aquellas soluciones de la región factible en las que todas o algunas de las variables de decisión toman valores enteros. Este tipo de problema se denominan en general de *programación lineal entera*. Si todas las variables del problema deben ser enteras se habla de *programación entera pura*, pero si sólo algunas deben ser enteras y las restantes continuas se habla de *programación lineal entera-mixta.* Cuando las variables enteras están restringidas a los dos valores 0-1, se denominan *variables binarias*, y el problema correspondiente *problema binario*.

Como veremos en el tema 6 de la asignatura, la resolución de los problemas enteros resulta más compleja que los continuos. En este caso la solución óptima no necesariamente tiene que coincidir con un vértice de la región factible del problema continuo, sino que puede estar en el interior o en las aristas de dicha región, pero siempre en puntos con valor entero de sus coordenadas.

Por ejemplo, el siguiente modelo lineal entero puro tiene la solución óptima en el punto (3, 2), bastante alejado del punto extremo donde tiene su valor óptimo el problema continuo.

$$
\begin{aligned}
&\textit{Minimizar } -6x_1 - 5x_2\\
&\textit{subject to}\\
&\qquad 3x_1 + x_2 \le 11\\
&\qquad -x_1 + 2x_2 \le 5\\
&\qquad x_1, x_2 \ge 0, \textit{integer}
\end{aligned}
$$



La utilización de variables enteras en general y binarias en particular amplía notablemente las posibilidades de modelado de la programación lineal, haciendo posible la disyunción de restricciones, la implicación lógica entre restricciones y en general la incorporación al modelo de ciertos comportamientos no lineales de la realidad.

En este tema exploraremos algunas de las nuevas posibilidades que introducen las variables enteras desde el punto de vista del modelado de problemas. También veremos su expresión en el leguaje OPL. En el tema 6 estudiaremos los métodos algorítmicos que se utilizan para resolverlos.

Dominios de aplicación de la Programación Entera

La programación entera resulta de interés en el modelado de los siguientes dominios de aplicación:

Aplicaciones con entrada y salida de datos discretos

Se trata de las aplicaciones en las que la programación entera se hace más evidente. Surgen cuando se quieren modelar plantas que fabrican productos con valor añadido muy alto y en un número de unidades entero y relativamente pequeño, por ejemplo, vehículos de transporte, equipos electrónicos de alta tecnología, etc. Por el contrario, si los valores enteros que se manejan en estos problemas son elevados, podrían resolverse como si fuesen de programación lineal continua y posteriormente redondear la solución.

Aplicaciones con relaciones lógicas entre variables o restricciones

Ocurre en bastantes ocasiones reales que es necesario establecer relaciones lógicas entre las restricciones que se deben imponer. Por ejemplo, .*"si se abre una fábrica en Zaragoza, se puede abrir también un almacén".* Este tipo de relaciones lógicas se pueden modelar introduciendo nuevas variables binarias.

Aplicaciones de optimización combinatoria

Muchos problemas prácticos de optimización tienen como característica básica la existencia de un número extremadamente grande de soluciones factibles. Dichas soluciones aparecen como consecuencia de diferentes métodos de ordenar actividades y asignar recursos. Este tipo de problemas se denominan combinatorios.

Linealización de problemas No-lineales

Como veremos en el tema 9 muchos problemas no-lineales se pueden aproximar utilizando modelos de programación entera binaria.

Problema 1

Se están considerando cuatro posibles inversiones. La primera de ellas se prevé que proporcione unos beneficios netos de 16.000 euros, la segunda, 22.000 euros, la tercera 12.000 euros, y la cuarta 8.000 euros. Cada una de las inversiones requiere una cantidad de dinero en efectivo: 5.000, 7.000, 4.000 y 3.000 euros, respectivamente. Si solo se dispone de 14.000 euros para invertir. ¿Qué modelo de programación lineal entera permite obtener la combinación de inversiones que prevea los máximos beneficios?

Solución

NUEVO
La declaración de variables enteras en OPL se realiza con la palabra clave int seguida del nombre de la variable y de la definición del rango de variación. El rango es obligatorio y se compone de la palabra clave in seguida del intervalo de variación:
extremo inferior..*extremo superior*

Variables de decisión

$$x_i = \begin{cases} 1 & \text{si se elige la inversión } i \\ 0 & \text{si no se elije} \end{cases} \quad i = 1,2,3,4$$

Restricciones

La suma de los costes de las inversiones no debe rebasar la cantidad total disponible.

$$5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14$$
$$x_i \in \{0,1\}, \quad i = 1,2,3,4$$

Función objetivo

Hay que maximizar el beneficio total de todas las inversiones

$$\text{Maximizar} \quad z = 16x_1 + 22x_2 + 12x_3 + 8x_4$$

Modelo OPL

```
dvar int x1 in 0..1;
dvar int x2 in 0..1;
dvar int x3 in 0..1;
dvar int x4 in 0..1;

maximize 16*x1+22*x2+12*x3+8*x4;

subject to
{
    5*x1+7*x2+4*x3+3*x4 <= 14;
}
```

```
// solution (optimal) with objective 42
x1 = 0;
x2 = 1;
x3 = 1;
x4 = 1;
```

Luego las inversiones segunda, tercera y cuarta producen el máximo beneficio de 42.000 euros

Problema 2

Se desea ampliar una compañía con la instalación de una nueva factoría en Zaragoza o Sevilla o en ambas ciudades. También se piensa construir a lo sumo un almacén en una ciudad donde se instale alguna factoría. En la siguiente tabla aparecen los beneficios estimados de instalar una factoría y construir un almacén en Zaragoza y Sevilla, y el capital requerido para ello. Se dispone de un capital total para la inversión de 39 M euros. El objetivo es tomar las decisiones que optimicen el beneficio de la inversión.

Decisiones	Beneficio estimado	Capital requerido
Instalar una factoría en Zaragoza	19 M euros	16 M euros
Instalar una factoría en Sevilla	15 M euros	13 M euros
Construir un almacén en Zaragoza	16 M euros	15 M euros
Construir un almacén en Sevilla	14 M euros	12 M euros

Solución

Variables de decisión

x_1 = *Instalar una factoría en Zaragoza*

x_2 = *Instalar una factoría en Sevilla*

x_3 = *Construir un almacén en Zaragoza*

x_4 = *Construir un almacén en Sevilla*

$$x_i = \begin{cases} 1 \text{ si la decisión i es sí} \\ 0 \text{ si la decisión i es no} \end{cases}$$

Restricciones

$$16x_1 + 13x_2 + 15x_3 + 12x_4 \leq 39$$

El conjunto de decisiones que se adopte no debe rebasar el presupuesto de inversión disponible

$$x_3 + x_4 \leq 1$$

Se debe construir a lo sumo un almacén

$$x_3 \leq x_1$$

El almacén en Zaragoza sólo se puede construir si se instala una factoría en Zaragoza

$$x_4 \leq x_2$$

El almacén en Sevilla sólo se puede construir si se instala una factoría en Sevilla

Función objetivo

Hay que maximizar el beneficio total de todas las decisiones de inversión

$$\text{Maximizar } Z = 19x_1 + 15x_2 + 16x_3 + 14x_4$$

Modelo OPL

```
dvar int x1 in 0..1;
dvar int x2 in 0..1;
dvar int x3 in 0..1;
dvar int x4 in 0..1;

maximize 19*x1+15*x2+16*x3+14*x4;

subject to
{
  16*x1+13*x2+15*x3+12*x4 <= 29;
  x3+x4 <= 1;
  x3 <= x1;
  x4 <= x2;
}
```

```
// solution (optimal) with objective 35

x1 = 1;
x2 = 0;
x3 = 1;
x4 = 0;
```

Solución optima: construir una factoría en Zaragoza y el almacén también en Zaragoza, con un beneficio estimado de 35 M euros.

Variables indicadoras: control del valor de una variable continua

Es posible establecer relaciones lógicas entre restricciones lineales usando variables enteras binarias y manteniendo la linealidad. En este apartado vamos a utilizar estas variables como indicadoras de que se cumple una condición en una variable continua. Por ejemplo, la variable binaria d podemos utilizarla para indicar con valor 1 que la variable continua x es mayor que cero, es decir:

$$x > 0 \rightarrow d = 1$$

Esta implicación lógica la podemos modelar con la siguiente restricción:

$$x - M \cdot d \leq 0$$

Siendo M una cota superior del valor de x. En efecto, cuando x es mayor que 0 necesariamente d tendrá que valer 1 para que el lado izquierdo de la restricción sea efectivamente menor o igual que 0.

La restricción anterior nos asegura que d tomará el valor 1 siempre que $x > 0$, pero no el sentido opuesto de esta implicación, es decir, que siempre que $d = 1$ se cumpla que $x > 0$. Para modelar en la práctica el sentido opuesto de esta implicación será necesario la definición de un cierto umbral m para x a partir del cual se considere que x deja de ser cero y toma un valor positivo, es decir:

$$d = 1 \rightarrow x > m$$

Esta implicación lógica la podemos modelar con la siguiente restricción:

$$x - m \cdot d \geq 0$$

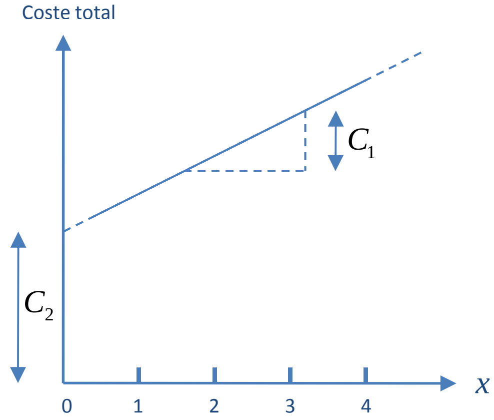
En algunas ocasiones no es necesario imponer explícitamente esta segunda restricción porque la condición de óptimo la impone implícitamente, como ocurre en el modelado de una función objetivo de coste fijo que analizamos a continuación.

Costes fijos

Las variables indicadoras se puede utilizar para modelar funciones de coste en las que aparece una componente de coste fijo. Se trata de casos en los que el coste total de una actividad es la suma de un coste variable, casi siempre proporcional al nivel de la actividad, y un coste fijo necesario para iniciarla (*set-up*). Si C_1 es el coste por unidad de producto y C_2 el coste de inicialización , el coste total vendrá dado por:

$$
\begin{aligned}
&Si\ x = 0 \rightarrow coste\ total = 0 \\
&Si\ x > 0 \rightarrow coste\ total = C_1 x + C_2
\end{aligned}
$$

La gráfica de la derecha recoge el comportamiento de estas funciones de coste.



El modelo utilizará una variable indicadora d y las restricciones que aseguran las dos condiciones anteriores sobre el valor de x:

$$
\begin{aligned}
&Minimizar\ z = C_1 \cdot x + C_2 \\
&sujeto\ a: \quad x - M \cdot d \le 0 \\
&\qquad\qquad\quad x - m \cdot d \ge 0
\end{aligned}
$$

En este caso podemos eliminar la segunda restricción ya que al tratarse de un problema de minimización, queda asegura por la condición de óptimo:

$$
\begin{aligned}
&Minimizar\ z = C_1 \cdot x + C_2 \\
&suejeto\ a: \ x - M \cdot d \le 0
\end{aligned}
$$

Problema 3: coste fijo

Tres compañías de teléfonos ofrecen su servicio de larga distancia con Estados Unidos en las siguientes condiciones:

- La compañía C1 cobra una tarifa fija de 16 euros al mes, más 0,25 céntimos por minuto.
- La compañía C2 cobra 25 euros al mes de tarifa fija, pero reduce el coste por minuto a 0,21 céntimos.
- La compañía C3 ofrece una tarifa fija mensual de 18 euros y un coste por minuto de 0,22 céntimos.

Las compañías sólo cobran la tarifa fija si se realiza alguna llamada a través de su operador. Teniendo en cuenta que un usuario consume un promedio mensual de 200 minutos en llamadas a Estados Unidos, y que puede repartir dichas llamadas entre las tres compañías, ¿qué servicios debe utilizar para que la factura mensual de teléfono sea lo más económica posible?

Solución

Variables de decisión

$$x_i = minutos\ mensuales\ consumidos\ con\ Ci$$

$$y_i = \begin{cases} 1\ si\ x_i > 0 \\ 0\ caso\ contrario \end{cases}$$

$$i = 1, 2, 3$$

200 es una cota superior de las tres variables de decisión continuas *xi*

Restricciones

$$x_1 - 200 y_1 \le 0$$

$$x_2 - 200 y_2 \le 0$$

$$x_3 - 200 y_3 \le 0$$

$$x_1, x_2, x_3 \ge 0$$

$$y_1, y_2, y_3 \in \{0,1\}$$

Hay que minimizar el coste total de la factura

Función objetivo

$$Minimizar\ \ z = 0,25x_1 + 0,21x_2 + 0,22x_3 + 16y_1 + 25y_2 + 18y_3$$

Modelo OPL

```
dvar float+ x1;
dvar float+ x2;
dvar float+ x3;
dvar int y1 in 0..1;
dvar int y2 in 0..1;
dvar int y3 in 0..1;

minimize 0.25*x1+0.21*x2+0.22*x3+
         16*y1+25*y2+18*y3;

subject to
{
   x1+x2+x3==200;
   x1-200*y1 <= 0;
   x2-200*y2 <= 0;
   x3-200*y3 <= 0;
}
```

```
// solution (optimal) with objective 62
x1 = 0;
x2 = 0;
x3 = 200;
y1 = 0;
y2 = 0;
y3 = 1;
```

Solución optima: contratar con la compañía C3, resultando una factura mensual de 62 euros.

Problema 4: recubrimiento

Se desea construir el menor número de estaciones de bomberos que cubra un territorio de 6 ciudades C1, C2, C3, C4, C5, y C6. Las estaciones se podrían construir en cualquiera de las ciudades pero garantizando siempre que todas las ciudades dispongan al menos de una estación a una de distancia máxima de 15 minutos. En la siguiente tabla se dan los tiempos en minutos para ir de una ciudad a otra:

	C1	C2	C3	C4	C5	C6
C1	0	10	20	30	30	20
C2	10	0	25	35	20	10
C3	20	25	0	15	30	20
C4	30	35	15	0	15	25
C5	30	20	30	15	0	14
C6	20	10	20	25	14	0

Se debe averiguar el número de estaciones de bomberos a construir y la ciudad donde deben construirse.

Solución

Variables de decisión

$$x_i = \begin{cases} 1 & si\ se\ elige\ la\ ciudad\ i \\ 0 & si\ no\ se\ elije \end{cases} \quad i = 1,2,3,4,5,6$$

Restricciones

$$x_1 + x_2 \geq 1$$
$$x_1 + x_2 + x_6 \geq 1$$
$$x_3 + x_4 \geq 1$$
$$x_3 + x_4 + x_5 \geq 1$$
$$x_4 + x_5 + x_6 \geq 1$$
$$x_2 + x_5 + x_6 \geq 1$$

	C1	C2	C3	C4	C5	C6
C1	0	10	20	30	30	20
C2	10	0	25	35	20	10
C3	20	25	0	15	30	20
C4	30	35	15	0	15	25
C5	30	20	30	15	0	14
C6	20	10	20	25	14	0

Para cada ciudad hay que imponer el recubrimiento con el resto de ciudades que se encuentran a 15 o menos minutos de ella (coloreadas y en la misma fila de la tabla)

Función de coste

El número de estaciones de bomberos debe ser mínimo

$$Minimizarz = x_1 + x_2 + x_3 + x_4 + x_5 + x_6$$

Modelo OPL

```
dvar int x1 in 0..1;
dvar int x2 in 0..1;
dvar int x3 in 0..1;
dvar int x4 in 0..1;
dvar int x5 in 0..1;
dvar int x6 in 0..1;

minimize x1+x2+x3+x4+x5+x6;

subject to
{
x1+x2 >=1;
x1+x2+x6 >=1;
x3+x4 >=1;
x3+x4+x5 >=1;
x4+x5+x6 >=1;
x2+x5+x6 >=1;
}
```

```
// solution (optimal) with objective 2
x1 = 0;
x2 = 1;
x3 = 0;
x4 = 1;
x5 = 0;
x6 = 0;
```

Solución optima: construir dos estaciones de bomberos en las ciudades C2 y C4

Modelado de la disyunción entre dos restricciones lineales utilizando variables binarias

Se trata de imponer el cumplimiento de una restricción entre dos restricciones, es decir, que se cumpla:

$$x_1 + 2x_2 + 2x_3 \le 30$$
$$\text{ó} \qquad\qquad x_1, x_2, x_3 \ge 0$$
$$2x_1 + x_2 + 2x_3 \le 45$$

Para ello tenemos en cuenta lo siguiente:

$$\textit{Si } M \textit{ es una cota superior de } x_1 + 2x_2 + 2x_3 \text{-}30$$
$$\textit{entonces } \; x_1 + 2x_2 + 2x_3 \le 30 + \mathrm{M} \quad \textit{se cumple } \forall x_1, x_2, x_3$$

En realidad la restricción anterior no impone ningún valor a sus variables fuera de su rango de variación, es decir, al sumarle la cota M al lado derecho anulamos la restricción. Por tanto introduciendo una variable binaria y, el siguiente programa hace que sólo se imponga una de las restricciones:

$$x_1 + 2x_2 + 2x_3 \le 30 + My$$
$$2x_1 + x_2 + 2x_3 \le 45 + M^{'}(1 - y)$$
$$x_1, x_2, x_3 \ge 0$$
$$\mathrm{y} \in \{0,1\}$$

Si $y = 1$ se anula esta restricción, si $y = 0$ se impone

Si $y = 1$ se impone esta restricción, si $y = 0$ se anula

Como y debe tomar uno de los valores 0 o 1, se impone necesariamente una de las dos restricciones

El mismo comportamiento se consigue introduciendo dos variables binarias e imponiendo que su suma valga 1:

$$x_1 + 2x_2 + 2x_3 \le 30 + My_1$$
$$2x_1 + x_2 + 2x_3 \le 45 + M^{'} y_2$$
$$\mathrm{y}_1 + y_2 = 1$$
$$x_1, x_2, x_3 \ge 0$$
$$\mathrm{y}_1, y_2 \in \{0,1\}$$

Los casos anteriores modelan la disyunción exclusiva de la dos restricciones, porque forzosamente se impone una y sólo una de las dos. En cambio, si imponemos que la suma de las dos variables binarias pueda ser menor que 1, es decir 0, admitimos la posibilidad de que no se imponga ninguna de las dos, es decir, que como máximo se cumple una de ellas:

$$x_1 + 2x_2 + 2x_3 \le 30 + My_1$$
$$2x_1 + x_2 + 2x_3 \le 45 + My_2$$
$$y_1 + y_2 \le 1$$
$$x_1, x_2, x_3 \ge 0$$
$$y_1, y_2 \in \{0,1\}$$

El caso anterior podemos generalizarlo al caso de *N* restricciones lineales de las cuales se imponen K < N:

$$f_1\ (x_1, x_2, ..., x_n) \le b_1$$
$$f_2\ (x_1, x_2, ..., x_n) \le b_2$$
$$.....$$
$$f_N\ (x_1, x_2, ..., x_n) \le b_N$$

Para ello introducimos N variables binarias y exigimos que su suma sea igual a *N* - *K*:

$$f_1\ (x_1, x_2, ..., x_n) \le b_1 + M_1 y_1$$
$$f_2\ (x_1, x_2, ..., x_n) \le b_2 + M_2 y_2$$
$$.....$$
$$f_N\ (x_1, x_2, ..., x_n) \le b_N + M_N y_N$$
$$\sum_{i=1}^{N} y_i = N - K$$
$$y_i \in \{0,1\}; \forall i = 1, ...N$$
$$M_i = cota\ superior\ de\ f_i\ (x_1, x_2, ..., x_n) - b_i$$

Problema 5

Una compañía está considerando la fabricación de tres tipos nuevos de vehículos: T1, T2, y T3. Los recursos necesarios para su fabricación, los recursos disponibles, y los beneficios esperados, para cada tipo de vehículo, se dan en la siguiente tabla:

Tipos	T1	T2	T3	Disponibilidad
Material	1500 kilos	3000 kilos	5000 kilos	6000000 kilos
Trabajo	30 horas	25 horas	40 horas	60000 horas
Beneficios	2000 euros	3000 euros	4000 euros	

La empresa quiere conocer qué tipo de vehículos debe fabricar y cuántos para maximizar los beneficios, teniendo en cuenta que un nuevo modelo solo resulta económicamente viable si se fabrican al menos 1000 unidades.

Solución

Variables de decisión

2.400 es una cota del número de vehículos que se pueden construir impuesta por las horas disponibles: 60.000/25 = 2.400 . Los kilos disponibles imponen una cota menos restrictiva, 4.000

$$x_i = número\ de\ vehículos\ a\ construir\ del\ tipo\ T_i, \quad i = 1,2,3$$

$$y_i = variable\ binaria\ auxiliar\ para\ expresar\ la\ disyunción$$

Restricciones

$$x_1 \le 2400 y_1$$

$$1000 - x_1 \le 2400(1 - y_1)$$

$$x_2 \le 2400 y_2$$

$$1000 - x_2 \le 2400(1 - y_2)$$

$$x_3 \le 2400 y_3$$

$$1000 - x_3 \le 2400(1 - y_3)$$

$$1,5x_1 + 3x_2 + 5x_3 \le 6000$$

$$30x_1 + 25x_2 + 40x_3 \le 60000$$

Si $y_1 = 0$ entonces $x_1 = 0$, luego no se fabrican vehículos del tipo T_1. Igual ocurre para *i=2 y 3*

Si $y_1 = 1$ entonces $x_1 >= 1000$, luego se fabrican vehículos del tipo T_1. Igual ocurre para *i=2 y 3*

Los kilos de material consumido no debe sobrepasar los kilos disponibles

Las horas de trabajo consumidas no debe sobrepasar los horas disponibles

Función de coste

Hay que maximizar el beneficio total

$$Maximizar\ z = 2x_1 + 3x_2 + 4x_3$$

Modelo OPL

```
dvar float+ x1;
dvar float+ x2;
dvar float+ x3;
dvar int y1 in 0..1;
dvar int y2 in 0..1;
dvar int y3 in 0..1;
maximize 2*x1+3*x2+4*x3;
subject to
{
x1 <= 2400 * y1;
1000 - x1 <= 2400 * (1 - y1);
x2 <= 2400 * y2;
1000 - x2 <= 2400 * (1 - y2);
x3 <= 2400 * y3;
1000 - x3 <= 2400 * (1 - y3);
1.5*x1+3*x2+5*x3 <= 6000;
30*x1+25*x2+40*x3 <= 60000;
}
```

Se podían haber usado variables int

```
// solution (optimal) with objective 6000
x1 = 0;
x2 = 2000;
x3 = 0;
y1 = 0;
y2 = 1;
y3 = 0;
```

Solución optima: construir 2000 vehículos de tipo T2 con un beneficio de 6.000.000

Problema de la mochila

El problema de la mochila es un problema clásico que puede formularse sólo con variables binarias 0-1. Toma su nombre de la versión que plantea la decisión que debe tomar un excursionista para preparar su mochila introduciendo una serie de objetos de utilidad, pero teniendo en cuenta que en la misma sólo caben un número limitado de ellos. Debe elegir un subconjunto de los objetos que maximice la utilidad total que obtiene, pero sin rebasar la capacidad de la mochila.

Si n es el número de objetos posibles, v_i el volumen que ocupa el objeto i, u_i el valor de utilidad que el excursionista da al objeto i, y b el volumen de la mochila, el problema se plantea de la siguiente forma:

$$Variables de decisión: x_i = \begin{cases} 1 \text{ si se elige el objeto } i \text{ para introducir en la mochila} \\ 0 \text{ caso contraio} \end{cases}$$

$$i = 1,2,\ldots,n$$

$$Maximizar\ z = \sum_{i=1}^{n} u_i x_i$$

Se maximiza la utilidad total de los objetos introducidos en la mochila, que es la suma de las utilidades de cada objeto

$$sujeto\ a:$$

$$\sum_{i=1}^{n} v_i x_i \leq b$$

$$x_i \in \{0,1\}$$

Los objetos introducidos no deben sobrepasar el volumen de la mochila

Podrían existir más dimensiones que limiten los objetos que se pueden introducir en la mochila, por ejemplo, el peso. Si p_i es el peso del objeto i, y p el peso total que puede soportar la mochila, habría que añadir en este caso la siguiente restricción;

$$\sum_{i=1}^{n} p_i x_i \leq p$$

Problema 6: mochila

Se debe realizar un envío de 7 objetos distintos. El valor, peso y volumen aparecen en la siguiente tabla:

Objeto	Valor (euros)	Peso (kilos)	Volumen (cm^3)
1	56	7	21
2	71	11	16
3	69	4	17
4	91	14	28
5	70	9	12
6	85	2	31
7	65	12	19

Determinar el envío de valor máximo que no exceda el peso total de 41 kilos ni el volumen de 100 cm^3.

Solución

Variables de decisión

$$x_i = \begin{cases} 1 \; si \; se \; incluye \; el \; objeto \; i \;\; en \; el \; envío \\ 0 \; caso \; contraio \end{cases}$$

$$i = 1, 2, 3, 4, 5, 6, 7$$

Los objetos introducidos no deben sobrepasar el peso de la mochila

Los objetos introducidos no deben sobrepasar el volumen de la mochila

Restricciones

$$7x_1 + 11x_2 + 4x_3 + 14x_4 + 9x_5 + 2x_6 + 12x_7 \leq 41$$

$$21x_1 + 16x_2 + 17x_3 + 28x_4 + 12x_5 + 31x_6 + 19x_7 \leq 100$$

Se maximiza el valor total de los objetos introducidos en la mochila.

Función de coste

$$Maximizar \;\; z = 56x_1 + 71x_2 + 69x_3 + 91x_4 + 70x_5 + 85x_6 + 65x_7$$

Modelo OPL

```
dvar int x1 in 0..1;
dvar int x2 in 0..1;
dvar int x3 in 0..1;
dvar int x4 in 0..1;
dvar int x5 in 0..1;
dvar int x6 in 0..1;
dvar int x7 in 0..1;
maximize 56*x1+71*x2+69*x3+91*x4+70*x5+85*x6+65*x7;
subject to
{
   7*x1+11*x2+4*x3+14*x4+9*x5+2*x6+12*x7 <=41;
   21*x1+16*x2+17*x3+28*x4+12*x5+31*x6+19*x7 <=100;
}
```

```
// solution (optimal) with objective 360
x1 = 0;
x2 = 1;
x3 = 1;
x4 = 0;
x5 = 1;
x6 = 1;
x7 = 1;
```

Solución optima: el envío estará constituido por los objetos 2, 3, 5, 6 y 7; con un valor de 360 euros

Problema 7: asignación de tareas a máquinas

Una fábrica realiza 3 tareas diferentes asociadas a la elaboración de otros tantos productos. La fábrica dispone de 4 máquinas que son utilizadas para la realización de las tareas. En la siguiente tabla aparece la secuencia de máquinas que utiliza cada tarea y el tiempo en minutos que la tarea ocupa en la correspondiente máquina. También aparece en la última columna el tiempo máximo que puede durar una tarea, desde que empieza hasta que termina. Cada máquina sólo puede realizar una tarea simultáneamente.

Tarea	Sucesión de máquinas (tiempo de ocupación de la máquina)	Tiempo máximo por tarea
T1	M1(4)→M3(3)→M4(5)	16
T2	M1(2)→M2(6)→M3(1)	14
T3	M2(7)→M4(4)	14

Se trata de diseñar un modelo lineal de programación de las 3 tareas sobre las 4 máquinas para que se realicen en el menor tiempo posible.

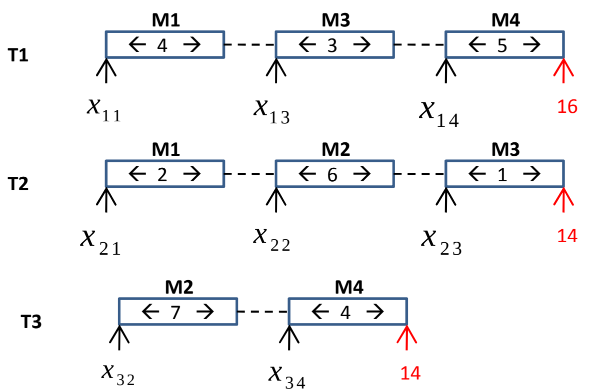
Solución

Variables de decisión

$$x_{ij} = \textit{instante de inicio de la tarea } T_i \textit{ en la máquina } M_j; \quad i = 1,2,3; \quad j = 1,2,3,4$$

El inicio de la tarea T_1 en la máquina M_3 tiene que comenzar cuando haya finalizado en la máquina M_1, es decir, a partir de 4 minutos después del inicio en M_1. Análogamente para las demás tareas.

Restricciones de secuenciación de cada tarea en las máquinas



$$x_{13} \geq x_{11} + 4 \qquad x_{14} \geq x_{13} + 3$$

$$x_{22} \geq x_{21} + 2 \qquad x_{23} \geq x_{22} + 6$$

$$x_{34} \geq x_{32} + 7$$

La tarea T_1 no debe durar más de 16 minutos, es decir, el inicio en la última máquina (M_4) más la duración en esta máquina (5) debe ser <= que 16. Análogamente para las demás tareas.

$$x_{14} + 5 - x_{11} \leq 16$$

$$x_{23} + 1 - x_{21} \leq 14$$

$$x_{34} + 4 - x_{32} \leq 14$$

Solución (continuación)

Restricciones de ordenación de las tareas en las mismas máquinas

$$x_{21} + 2 - x_{11} \le Md_1$$
$$x_{11} + 4 - x_{21} \le M(1 - d_1)$$

Las tareas T_1 y T_2 utilizan la máquina M_1, por tanto hay que imponer la disyunción exclusiva del inicio de ambas tareas en esta máquina. Para ello utilizamos la variable binaria d_1.

$$x_{23} + 1 - x_{13} \le Md_2$$
$$x_{13} + 3 - x_{23} \le M(1 - d_2)$$

Las tareas T_1 y T_2 utilizan la máquina M_3, por tanto hay que imponer la disyunción exclusiva del inicio de ambas tareas en esta máquina. Para ello utilizamos la variable binaria d_2.

$$x_{34} + 4 - x_{14} \le Md_3$$
$$x_{14} + 5 - x_{34} \le M(1 - d_3)$$

Las tareas T_1 y T_3 utilizan la máquina M_4, por tanto hay que imponer la disyunción exclusiva del inicio de ambas tareas en esta máquina. Para ello utilizamos la variable binaria d_3.

$$x_{32} + 7 - x_{22} \le Md_4$$
$$x_{22} + 6 - x_{32} \le M(1 - d_4)$$

Las tareas T_3 y T_2 utilizan la máquina M_2, por tanto hay que imponer la disyunción exclusiva del inicio de ambas tareas en esta máquina. Para ello utilizamos la variable binaria d_4.

Función objetivo

$$Minimizar \quad t$$

$$x_{14} + 5 \le t$$
$$x_{23} + 1 \le t$$
$$x_{34} + 4 \le t$$

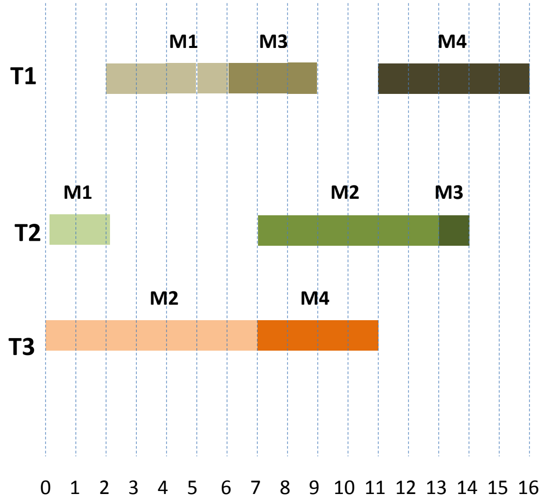
Para expresar la función de coste que minimiza el tiempo de finalización de todas las tareas introducimos una nueva variable t y le imponemos que sea mayor que el tiempo de finalización de las tres tareas (tiempo de inicio en la última máquina utilizada + duración en esa máquina). Después minimizamos t.

Problema 7: modelo OPL y resultados

Modelo OPL

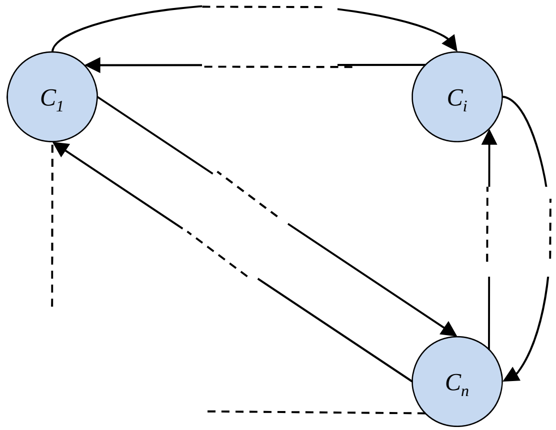
```
dvar int x11 in 0..100;
dvar int x13 in 0..100;
dvar int x14 in 0..100;
dvar int x21 in 0..100;
dvar int x22 in 0..100;
dvar int x23 in 0..100;
dvar int x32 in 0..100;
dvar int x34 in 0..100;
dvar int d1 in 0..1;
dvar int d2 in 0..1;
dvar int d3 in 0..1;
dvar int d4 in 0..1;
dvar int t in 0..100;
minimize t;
subject to
{
x13 >= x11 + 4;  x14 >= x13 + 3;
x22 >= x21 + 2;  x23 >= x22 + 6;
x34 >= x32 + 7;
x14 + 5 – x11 <= 16;
x23 + 1 – x21 <= 14;
x34 + 4 – x32 <= 14;
x21 + 2 - x11 <= 100*d1;
x11 + 4 - x21 <= 100*(1-d1);
x23 + 1 - x13 <= 100*d2;
x13 + 3 - x23 <= 100*(1-d2);
x34 + 4 - x14 <= 100*d3;
x14 + 5 - x34 <= 100*(1-d3);
x32 + 7 - x22 <= 100*d4;
x22 + 6 - x32 <= 100*(1-d4);
x14 + 5 <= t;
x23 + 1 <= t;
x34 + 4 <= t;
}
```

```
// solution (optimal) with objective 16
t = 16;
x13 = 8;
x11 = 2;
x14 = 11;
x22 = 7;
x21 = 0;
x23 = 13;
x34 = 7;
x32 = 0;
d1 = 0;
d2 = 1;
d3 = 0;
d4 = 0;
```



Problema del viajante

Sobre una red de carreteras que conecta n ciudades $C_1, \dots ,C_i, \dots, C_n$ hay que determinar el itinerario de mínima distancia que partiendo de una ciudad pasa una sola vez por todas y cada una de las ciudades volviendo a la ciudad inicial.



Variables de decisión:

$$x_{ij} = \begin{cases} 1 \text{ si en el itinerario está el arco } C_i \to C_j \\ 0 \text{ caso contrario} \end{cases} \quad i = 1,\dots,n; \quad j = 1,\dots,n; \quad i \neq j$$

$$u_i = \text{variables auxiliares para evitar sub-itinerarios}; \quad i = 2,3,\dots,n$$

De una ciudad se parte una sola vez

$$\text{sujeto a}: \sum_{i=1}^{n} x_{ij} = 1 \quad i = 1,\dots n; \ i \neq j$$

A una ciudad se llega una sola vez

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

Evita sub-itinerarios desconectados
$Si\ x_{ij} = 1 \to u_i - u_j + n \leq n-1 \to u_i + 1 \leq u_j \to u_i < u_j \to x_{ji} = 0$

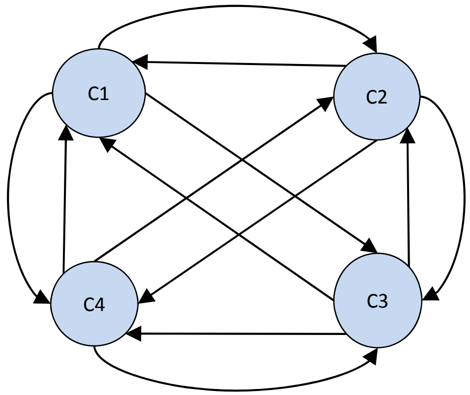
$$u_i - u_j + n \cdot x_{ij} \leq n-1; \quad i, j = 2,3,\dots,n; \quad i \neq j$$

Minimiza la distancia de a ruta

$$Minimizar\ L = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij}; \quad i \neq j$$

Problema 8: recorrido del viajante

Sobre una red de carreteras que conecta varias ciudades hay que determinar el itinerario de mínima distancia que partiendo de una ciudad pasa una sola vez por todas y cada una de ciudades volviendo a la ciudad inicial. Las distancias entre ciudades son las siguientes:



$$\begin{array}{ll} d_{12}=2 & d_{23}=4 \\ d_{21}=3 & d_{32}=5 \\ d_{13}=6 & d_{24}=2 \\ d_{31}=4 & d_{42}=3 \\ d_{14}=4 & d_{34}=5 \\ d_{41}=2 & d_{43}=3 \end{array}$$

Solución

Variables de decisión

$$x_{ij}=\begin{cases} 1 \; si \; en \; el \; itinerario \; está \; el \; arco \; C_i \to C_j \\ 0 \; caso \; contrario \end{cases} \qquad i=1,2,3,4; \qquad j=1,2,3,4; \qquad i \neq j$$

Restricciones

De una ciudad se parte una sola vez

A una ciudad se llega una sola vez

Evita sub-itinerarios desconectados

$$\begin{aligned} x_{12}+x_{13}+x_{14}&=1 \\ x_{21}+x_{23}+x_{24}&=1 \\ x_{31}+x_{32}+x_{34}&=1 \\ x_{41}+x_{42}+x_{43}&=1 \end{aligned}$$

$$\begin{aligned} x_{21}+x_{31}+x_{41}&=1 \\ x_{12}+x_{32}+x_{42}&=1 \\ x_{13}+x_{23}+x_{43}&=1 \\ x_{14}+x_{24}+x_{34}&=1 \end{aligned}$$

$$\begin{array}{ll} u_2 \le u_3 - 4x_{23}+3 & u_3 \le u_4 - 4x_{34}+3 \\ u_2 \le u_4 - 4x_{24}+3 & u_4 \le u_2 - 4x_{42}+3 \\ u_3 \le u_2 - 4x_{32}+3 & u_4 \le u_3 - 4x_{43}+3 \end{array}$$

Minimiza la distancia de a ruta

Función objetivo

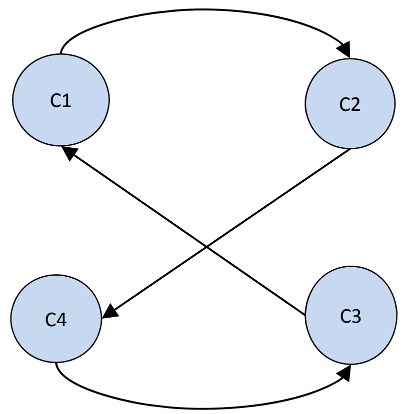
$$Minimizar \; z = 2x_{12}+3x_{21}+6x_{13}+4x_{31}+4x_{14}+2x_{41}+4x_{23}+5x_{32}+2x_{24}+3x_{42}+5x_{34}+3x_{43}$$

Problema 8: modelo OPL

Modelo OPL

```
dvar int x12 in 0..1;
dvar int x13 in 0..1;
dvar int x14 in 0..1;
dvar int x21 in 0..1;
dvar int x31 in 0..1;
dvar int x41 in 0..1;
dvar int x23 in 0..1;
dvar int x24 in 0..1;
dvar int x32 in 0..1;
dvar int x42 in 0..1;
dvar int x34 in 0..1;
dvar int x43 in 0..1;

dvar int u2 in 0..100;
dvar int u3 in 0..100;
dvar int u4 in 0..100;

minimize
2*x12+3*x21+6*x13+4*x31+4*x14+2*x41+4*x23+
5*x32+2*x24+3*x42+5*x34+3*x43;

subject to
{
x12+x13+x14 == 1;  x21+x31+x41 == 1;
x21+x23+x24 == 1;  x12+x32+x42 == 1;
x31+x32+x34 == 1;  x13+x23+x43 == 1;
x41+x42+x43 == 1;  x14+x24+x34 == 1;

u2 <= u3 -4*x23+3;  u3 <= u4 -4*x34+3;
u2 <= u4 -4*x24+3;  u4 <= u2 -4*x42+3;
u3 <= u2 -4*x32+3;  u4 <= u3 -4*x43+3;
}
```

```
// solution (optimal) with objective 11
x12 = 1;
x21 = 0;
x13 = 0;
x31 = 1;
x14 = 0;
x41 = 0;
x23 = 0;
x32 = 0;
x24 = 1;
x42 = 0;
x34 = 0;
x43 = 1;
u2 = 0;
u3 = 2;
u4 = 1;
```



Funciones con N valores posibles

Una función que deba tomar un valor v_i entre N valores posibles se puede modelar con la ayuda de N variables binarias y_i. En efecto, si la función es:

$$f(x_1, x_2, ..., x_n) = v_1 \quad \vee \quad v_2 \quad \vee \; ... \; \vee \quad v_N$$

Se modela con el siguiente conjunto de restricciones:

$$f(x_1, x_2, ..., x_n) = \sum_{i=1}^{N} v_i y_i$$

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

$$y_i \in \{0,1\}; \forall i = 1, ... N$$

Conversión de variables enteras en binarias

Como tendremos ocasión de estudiar en el tema 6 los algoritmos que resuelven problemas enteros binarios puros son más eficientes que los que involucran variables enteras en general. Por ello resulta interesante convertir variables enteras acotadas en variables binarias.

$$Si\ \ 0 \le x \le u;\ con\ x, u \in Z$$

$$podemos\ poner\ \ 2^N \le u \le 2^{N+1};\ con\ N \in Z$$

$$y\ la\ representación\ binaria\ de\ x\ será:\ x = \sum_{i=0}^{N} 2^i y_i;\ con\ y_i\ \ variables\ binarias$$

Ejemplo

$$\begin{aligned} &Maximizar\ \ z=x_1 + 2x_2 \\ &sujeto\ a \\ &\qquad\qquad x_1 \le 5 \\ &\qquad\qquad 2x_1 + 3x_2 \le 30 \\ &con\ x_1, x_2 \in Z \end{aligned}$$

$$x_1 \le 5 \qquad \text{cota superior de } x_1 = 5 \ \rightarrow 2^2 \le 5 \le 2^3 \ \rightarrow N = 2 \rightarrow x_1 = y_0 + 2y_1 + 4y_2$$

$$2x_1 + 3x_2 \le 30 \qquad \text{cota superior de } x_2 = 10 \rightarrow 2^3 \le 10 \le 2^4 \rightarrow N = 3 \rightarrow x_2 = y_3 + 2y_4 + 4y_5 + 8y_6$$

$$\begin{aligned} &Maximizar\ \ z=y_0 + 2y_1 + 4y_2 + y_3 + 4y_4 + 8y_5 + 16y_6 \\ &sujeto\ a \\ &\qquad\qquad y_0 + 2y_1 + 4y_2 \le 5 \\ &\qquad\qquad 2y_0 + 4y_1 + 8y_2 + 3y_3 + 6y_4 + 12y_5 + 24y_6 \le 30 \\ &con\ y_i \in \{0,1\} \end{aligned}$$

Problema:

```
dvar int y0 in 0..1;
dvar int y1 in 0..1;
dvar int y2 in 0..1;
dvar int y3 in 0..1;
dvar int y4 in 0..1;
dvar int y5 in 0..1;
dvar int y6 in 0..1;

maximize y0+2*y1+4*y2+y3+4*y4+8*y5+16*y6;

subject to
{
  y0+2*y1+4*y2<=5;
  2*y0+4*y1+8*y2+3*y3+6*y4+12*y5+24*y6<=30;
};
```

Modelo OPL (variables binarias)

```
dvar int x1 in 0..100;
dvar int x2 in 0..100;
maximize x1+2*x2;
subject to
{
  x1<= 5;
  2*x1+3*x2<=30;
};
```

Modelo OPL (variables enteras)

// solution (optimal) with objective 20

x1 = 0;
x2 = 10;

// solution (optimal) with objective 20

y0 = 0;
y1 = 0;
y2 = 0;
y3 = 0;
y4 = 1;
y5 = 0;
y6 = 1;

$$x_1 = y_0 + 2y_1 + 4y_2 = 0 + 2\times 0 + 4\times 0 = 0$$
$$x_2 = y_3 + 2y_4 + 4y_5 + 8y_6 = 0 + 2\times 1 + 4\times 0 + 8\times 1 = 10$$

Para determinar los valores enteros de las variables de decisión originales tenemos que utilizar sus expresiones en términos de las variables binarias:

$$x_1 = y_0 + 2y_1 + 4y_2$$
$$x_2 = y_3 + 2y_4 + 4y_5 + 8y_6$$

Expresión del producto de dos variables binarias

El producto de dos variables binarias $b_1 \cdot b_2$ es un término no lineal y por tanto no se puede utilizar directamente en el modelado de un problema de programación lineal. Sin embargo podemos expresar ese producto en términos de restricciones lineales, sustituyendo el producto por la variable binaria y junto al siguiente conjunto de restricciones lineales:

$$b_1 \cdot b_2 \Rightarrow \begin{cases} y \le b_1 \\ y \le b_2 \\ y \ge b_1 + b_2 - 1 \end{cases}$$

$$b_1, b_2, y \in \{0,1\}$$

En efecto, la variable introducida debe cumplir la siguiente tabla en función de las variables originales:

b_1	b_2	y
0	0	0
0	1	0
1	0	0
1	1	1

Tabla cuyo cumplimiento viene asegurado por las restricciones introducidas:

$y \le b_1 \leftrightarrow Si\ b_1 = 0 \Rightarrow y = 0$ — Asegura el cumplimiento de las entradas 1ª y 2ª de la tabla

$y \le b_2 \leftrightarrow Si\ b_2 = 0 \Rightarrow y = 0$ — Asegura el cumplimiento de las entradas 1ª y 3ª de la tabla

$y \ge b_1 + b_2 - 1 \leftrightarrow Si\ b_1 = 1 \wedge b_2 = 1 \Rightarrow y \ge 1 \Rightarrow y = 1$ — Asegura el cumplimiento de las entradas 4ª de la tabla

Expresión del producto de una variable binaria por otra continua no negativa

Si una de las variables del producto es continua, también podemos expresarlo en términos de restricciones lineales con variables binarias:

$$b \text{ variable binaria}$$
$$x \text{ variable continua con } 0 \le x \le u = \text{cota superior de } x$$

Podemos sustituir el producto por la variable y junto al siguiente conjunto de restricciones:

$$b \cdot x \Rightarrow \begin{cases} y \le ub \\ y \le x \\ y \ge x - u(1-b) \\ y \ge 0 \end{cases}$$
$$\text{con } y \text{ variable continua}$$

En efecto, la variable introducida debe cumplir la siguiente tabla en función de las variables originales:

b	*x*	*y*
0	*x*	*0*
1	*x*	*x*

Tabla cuyo cumplimiento viene asegurado por las restricciones introducidas:

Asegura el cumplimiento de la entrada 1ª de la tabla

b = 0 anula esta restricción porque x-u es siempre 0 o negativo

$$\left.\begin{array}{r} y \le ub \leftrightarrow Si\ b = 0 \Rightarrow y \le 0 \\ y \ge 0 \end{array}\right\} \Rightarrow y = 0; \qquad y \ge x - u(1-b) \Rightarrow y \ge x - u$$

Asegura el cumplimiento de la entrada 2ª de la tabla

b = 1 anula esta restricción porque y es menor que la cota de x

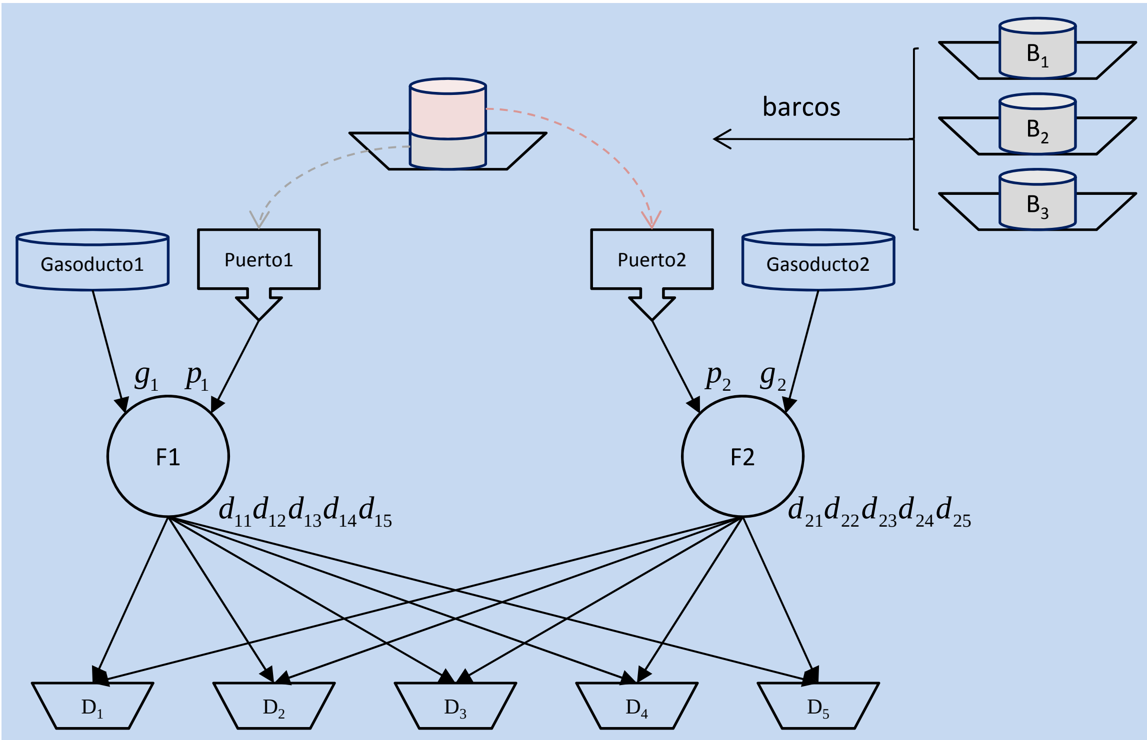
$$\left.\begin{array}{r} y \ge x - u(1-b) \leftrightarrow Si\ b = 1 \Rightarrow y \ge x \\ y \le x \end{array}\right\} \Rightarrow y = x; \quad y \le ub \Rightarrow y \le u$$

Problema: barcos a doble puerto

Dos factorías de gas F1 y F2 son suministradas desde gasoductos y barcos. Para un mes determinado existen tres barcos alternativos para suministrar gas con capacidades B_1, B_2, B_3 y costes respectivos del gas y transporte C_1, C_2, C_3. Los barcos pueden descargar a doble puerto, es decir, dejar parte de la carga en el Puerto1 y el resto en el Puerto2. Las factorías suministran gas a cinco puntos con demandas D_1,D_2,D_3,D_4,D_5 respectivamente, siendo C_{ij} el coste de suministrar desde la factoría i la demanda j. El coste unitario del gas procedente de gasoductos es G_1 y G_2 respectivamente. Se trata de determinar la cantidad de gas suministrada por gasoducto y barco a las factorías así como las cantidades transportadas desde las factorías a los puntos de demanda de manera que el coste total de suministro sea mínimo. En la siguiente figura aparecen los valores de las demandas, coste de transporte unitario de factorías a puntos de demanda, el coste unitario del gas procedente de gasoductos, y el coste de los barcos y sus capacidades. En el caso de suministro con barco a doble puerto habrá que determinar también la cantidad que descarga en cada puerto.

$C_{11} = 1$
$C_{12} = 1$
$C_{13} = 100$
$C_{14} = 100$
$C_{15} = 100$

$D_1 = 1$
$D_2 = 1,5$
$D_3 = 1$
$D_4 = 1$
$D_5 = 1$



$B_1 = 3$
$B_2 = 4$
$B_3 = 5$

$C_1 = 1$
$C_2 = 1$
$C_3 = 1$

$G_1 = 100$
$G_2 = 100$

$C_{21} = 100$
$C_{22} = 100$
$C_{23} = 1$
$C_{24} = 1$
$C_{25} = 1$

Modelo lineal

Variables de decisión:

$g_i =$ cantidad de gas suministrado por el gasoducto i; $i = 1,2$

$p_i =$ cantidad de gas suministrado por el puerto i; $i = 1,2$

$d_{ij} =$ gas suministrado por la factoría i al punto de demanda j; $i = 1,2$; $j = 1,2,3,4,5$

$b_i =$ variables binarias que determinan con valor 1 la selección del barco B_i; $i = 1,2,3$

$d_i =$ variables continuas $[0,1]$ que fija porcentajes de descarga del barco B_i; $i = 1,2,3$

$y_i =$ variables continuas para formular el producto $b_i \cdot d_i$; $i = 1,2,3$

Se minimiza la suma de todos los costes

$$\text{Minimizar } z = \sum_{i=1}^{2}\sum_{j=1}^{5} C_{ij} d_{ij} + \sum_{k=1}^{3} b_k C_k + \sum_{l=1}^{2} G_l g_l$$

sujeto a:

$$p_1 = b_1 B_1 d_1 + b_2 B_2 d_2 + b_3 B_3 d_3$$
$$p_2 = b_1 B_1 (1-d_1) + b_2 B_2 (1-d_2) + b_3 B_3 (1-d_3)$$
$$g_1 + p_1 = d_{11} + d_{12} + d_{13} + d_{14} + d_{15}$$
$$g_2 + p_2 = d_{21} + d_{22} + d_{23} + d_{24} + d_{25}$$
$$d_{11} + d_{21} = D_1$$
$$d_{12} + d_{22} = D_2$$
$$d_{13} + d_{23} = D_3$$
$$d_{14} + d_{24} = D_4$$
$$d_{15} + d_{25} = D_5$$

Los productos $b_i \cdot d_i$ se linealizarán introduciendo y_i

$$b_i \cdot d_i \Rightarrow \begin{cases} y_i \le b \\ y_i \le d_i \\ y_i \ge d_i - (1-b_i) \\ y_i \ge 0 \end{cases}$$

$$\text{Minimizar } z = \sum_{i=1}^{2}\sum_{j=1}^{5} C_{ij} d_{ij} + \sum_{k=1}^{3} b_k C_k + \sum_{l=1}^{2} G_l g_l$$

sujeto a:

$$p_1 = B_1 y_1 + B_2 y_2 + B_3 y_3$$
$$p_2 = B_1(b_1 - y_1) + B_2(b_2 - y_2) + B_3(b_3 - y_3)$$
$$g_1 + p_1 = d_{11} + d_{12} + d_{13} + d_{14} + d_{15}$$
$$g_2 + p_2 = d_{21} + d_{22} + d_{23} + d_{24} + d_{25}$$
$$d_{11} + d_{21} = D_1$$
$$d_{12} + d_{22} = D_2$$
$$d_{13} + d_{23} = D_3$$
$$d_{14} + d_{24} = D_4$$
$$d_{15} + d_{25} = D_5$$
$$y_1 \le b_1$$
$$y_1 \le d_1$$
$$y_1 \ge d_1 - (1-b_1)$$
$$y_1 \ge 0$$
$$y_2 \le b_2$$
$$y_2 \le d_2$$
$$y_2 \ge d_2 - (1-b_2)$$
$$y_2 \ge 0$$
$$y_3 \le b_3$$
$$y_3 \le d_3$$
$$y_3 \ge d_3 - (1-b_3)$$
$$y_3 \ge 0$$

Modelo OPL

```
//Variables de decisión
dvar int b1 in 0..1;
dvar int b2 in 0..1;
dvar int b3 in 0..1;

dvar float+ d1 in 0..1;
dvar float+ d2 in 0..1;
dvar float+ d3 in 0..1;

dvar float+ y1;
dvar float+ y2;
dvar float+ y3;

dvar float+ d11;
dvar float+ d12;
dvar float+ d13;
dvar float+ d14;
dvar float+ d15;

dvar float+ d21;
dvar float+ d22;
dvar float+ d23;
dvar float+ d24;
dvar float+ d25;

dvar float+ g1 in 0..1;
dvar float+ g2 in 0..1;

dvar float+ p1;
dvar float+ p2;
```

```
//Función objetivo
minimize 1*d11 + 1*d12 +100*d13 +100*d14 +100*d15 +
         100*d21 + 100*d22 +1*d23 +1*d24 +1*d25 +
         1*b1 + 1*b2 +1*b3 +100*g1 +100*g2;

//Restricciones
subject to
{
   p1==3*y1+4*y2+5*y3;
   p2==3*(b1-y1)+4*(b2-y2)+5*(b3-y3);
   g1+p1==d11+d12+d13+d14+d15;
   g2+p2==d21+d22+d23+d24+d25;
   d11+d21==1;
   d12+d22==1.5;
   d13+d23==1;
   d14+d24==1;
   d15+d25==1;
   y1<=b1;
   y1<=d1;
   y1>=d1-(1-b1);
   y2<=b2;
   y2<=d2;
   y2>=d2-(1-b2);
   y3<=b3;
   y3<=d3;
   y3>=d3-(1-b3);

  }
```

```
// solution (optimal) with objective
56.5
d11 = 1;
d12 = 1.5;
d13 = 0;
d14 = 0;
d15 = 0;
d21 = 0;
d22 = 0;
d23 = 1;
d24 = 1;
d25 = 1;
b1 = 0;
b2 = 0;
b3 = 1;
g1 = 0.5;
g2 = 0;
p1 = 2;
y1 = 0;
y2 = 0;
y3 = 0.4;
p2 = 3;
d1 = 0;
d2 = 0;
d3 = 0.4;
```

La Factoría1 suministra a las demandas 1 y 2

La Factoría2 suministra a las demandas 3, 4 y 5

Selecciona barco B_3

40% de B_3 descarga en Puerto1, el 60% en Puerto2

Variables indicadoras: control de la imposición de una restricción lineal

De la misma manera que hemos utilizado variables binarias indicadoras del valor de una variable continua, podemos utilizarlas como indicadoras del cumplimiento de una restricción lineal. En efecto, veamos las 4 alternativas que se presentan según el sentido de la implicación y del operador relacional de la restricción:

$d = 1 \rightarrow \sum_{i=1}^{n} a_i x_i \leq b$	$\sum_{i=1}^{n} a_i x_i + Md \leq M + b$ $siendo\ M \geq \sum_{i=1}^{n} a_i x_i - b$ Si $d = 1$ nos queda $\sum_{i=1}^{n} a_i x_i + \leq b$ Si $d = 0$ nos queda $\sum_{i=1}^{n} a_i x_i + \leq M + b$ que no restringe nada porque M es una cota superior de $\sum_{i=1}^{n} a_i x_i - b$
$\sum_{i=1}^{n} a_i x_i \leq b \rightarrow d = 1$	$\sum_{i=1}^{n} a_i x_i - (m - e)d \geq b + e; \quad siendo\ m \leq \sum_{i=1}^{n} a_i x_i - b$ $e = umbral\ a\ partir\ del\ cual\ consideramos\ \sum_{i=1}^{n} a_i x_i > b$
$d = 1 \rightarrow \sum_{i=1}^{n} a_i x_i \geq b$	$\sum_{i=1}^{n} a_i x_i + m \cdot d \geq m + b$ $siendo\ m \leq \sum_{i=1}^{n} a_i x_i - b$ Si $d = 1$ nos queda $\sum_{i=1}^{n} a_i x_i + \geq b$ Si $d = 0$ nos queda $\sum_{i=1}^{n} a_i x_i + \geq m + b$ que no restringe nada porque m es una cota inferior de $\sum_{i=1}^{n} a_i x_i - b$
$\sum_{i=1}^{n} a_i x_i \geq b \rightarrow d = 1$	$\sum_{i=1}^{n} a_i x_i - (M + e) \cdot d \geq b - e; \quad siendo\ M \geq \sum_{i=1}^{n} a_i x_i - b$ $e = umbral\ a\ partir\ del\ cual\ consideramos\ \sum_{i=1}^{n} a_i x_i < b$

Representación de operadores lógicos

Si la restricción R_i se cumple si y solo si $\delta_i = 1$, es decir $R_i \leftrightarrow \delta_i = 1$, entonces podemos escribir las siguientes equivalencias entre las relaciones lógicas entre restricciones y las operaciones algebraicas entre las variables indicadoras:

$$R_1 \vee R_2 \leftrightarrow \delta_1 + \delta_2 \geq 1$$
$$R_1 \cdot R_2 \leftrightarrow \delta_1 = 1 \; y \; \delta_2 = 1$$
$$\neg R_1 \leftrightarrow \delta_1 = 0$$
$$R_1 \rightarrow R_2 \leftrightarrow \delta_1 - \delta_2 \leq 0$$
$$X_1 \leftrightarrow X_2 \Leftrightarrow \delta_1 - \delta_2 = 0$$

1. Utilizando las relaciones de la lógica proposicional podemos obtener la expresión algebraica de las variables indicadoras correspondiente a cualquier expresión lógica entre restricciones.
2. Se supone que las variables indicadoras ya han sido introducidas con las restricciones analizadas en el punto anterior.

Ejemplo : conversión de una relación lógica en restricción

Suponga que se quiere garantizar que si $x \leq b$ y $y \geq 1$ entonces $y = z + 1$ Siendo x, y, z variables enteras, es decir:

$$(x \leq b) \wedge (x \geq 1) \rightarrow (y = z + 1)$$

$$(x \leq b) \wedge (x \geq 1) \rightarrow (\pi = 1) \rightarrow (y = z + 1)$$

$(x \leq b) \rightarrow (\alpha = 1)$	$x + M \cdot \alpha \geq b$
$(x \geq 1) \rightarrow (\beta = 1)$	$x - M \cdot \beta \leq 1$
$(\alpha = 1) \wedge (\beta = 1) \rightarrow (\pi = 1)$	$\alpha + \beta - \pi \leq 1$
$(\pi = 1) \rightarrow (y \geq z + 1)$	$y - z - m \cdot \pi \geq m + 1$
$(\pi = 1) \rightarrow (y \leq z + 1)$	$y - z + M \cdot \pi \leq M + 1$

$$donde: \;\; m \leq y - z - 1; \quad M \geq y - z + 1$$

Ejemplo

En un caso de programación de producción, si cualquiera de los productos A o B se fabrican, entonces hay que fabricar C, D, E.

$$(x_A \vee x_B) \rightarrow (x_C \vee x_D \vee x_E)$$

$$Sea \;\; \delta_i = 1 \leftrightarrow se \; fabrica \; el \; producto \; i$$

$$Introducimos \; \delta \; tal \; que: \; \delta_A + \delta_B \geq 1 \rightarrow \delta = 1, \quad que \; se \; puede \; modelar \; como: \boxed{\delta_A + \delta_B + 2\delta \leq 0}$$

$$Además: \quad \delta = 1 \rightarrow \delta_C + \delta_D + \delta_E \geq 1, \; que \; se \; puede \; modelar \; como: \boxed{\delta - \delta_C - \delta_D - \delta_E \leq 0}$$

Ejercicio

Una compañía está planificando la apertura de varias tiendas en una ciudad dividida en 7 distritos. Después de un estudio se determinaron los 5 emplazamientos posibles para las tiendas. La siguiente tabla muestra : a) los distritos cubiertos por cada emplazamiento, b) los beneficios esperados por cada tienda en los 5 emplazamientos, c) los costes de alquiler de los locales correspondientes a los 5 emplazamientos.

Distrito	Emplazamiento				
	1	2	3	4	5
1					
2					
3					
4					
5					
6					
7					
Beneficio(*10^4 euros)	36	39	44	41	38
Coste alquiler(*10^4 euros)	14	17	21	15	18

Determinar:

1) El emplazamiento de las tiendas que produce mayor beneficio de manera que cada distrito se cubra a lo sumo por una tienda.
2) El emplazamiento de las tiendas que minimice el coste de los alquileres aunque un distrito pueda cubrirse por más de una tienda.

Ejercicio alternativo

Al igual que hicimos en el tema 1, los alumnos pueden optar por definir su propio problema de programación entera con las siguientes fases:

1. Especificación de un problema real que pueda resolverse con un modelo de programación lineal entera (pura o mixta).
2. Diseño del modelo lineal entero correspondiente al problema especificado.
3. Expresión del modelo en OPL y ejecución en el entorno de desarrollo.