Programación Lineal

Luisa Carpente e Ignacio García Jurado

Departamento de Matemáticas
Universidade da Coruña

¿Qué es la investigación operativa?

- Una aproximación científica a la solución de problemas que surgen en la gestión de sistemas complejos (EURO).
- Es la disciplina científica que aplica métodos analíticos avanzados para ayudar a tomar mejores decisiones (INFORMS).
- La investigación operativa surge en la Segunda Guerra Mundial, cuando en los ejércitos aliados se reúne a grupos de científicos para analizar las operaciones militares. Después de la guerra comienza a aplicarse en el ámbito civil.
- La herramienta matemática principal de la investigación operativa son las técnicas de programación matemática (o de optimización). Otras herramientas de la investigación operativa son la teoría de la decisión, la teoría de la probabilidad y los métodos estadísticos.

¿Qué es un problema de programación matemática?

- Dada una función $f : \mathbb{R}^n \to \mathbb{R}$ y un conjunto $S \subset \mathbb{R}^n$, se trata de encontrar $\bar{x} \in \mathbb{R}^n$ que resuelva

$$\begin{array}{ll} max & f(x) \\ s.a: & x \in S. \end{array}$$

- Diremos que f es la *función objetivo* y que S es el *conjunto factible*.
- Si $\bar{x} \in \mathbb{R}^n$ diremos que $\bar{x}$ es una *solución*. Si, además, $\bar{x} \in S$ diremos que $\bar{x}$ es una *solución factible*. Si $\bar{x} \in S$ y $f(\bar{x}) \geq f(x)$ para todo $x \in S$ diremos que $\bar{x}$ es una *solución óptima*.
- Si $S = \emptyset$ diremos que el problema es *infactible*. Si existe una sucesión $\{x_k\} \subset S$ tal que $\{f(x_k)\} \to \infty$ cuando $k \to \infty$ diremos que el problema es *no acotado*. En los demás casos, es posible que el problema tenga soluciones óptimas.

¿Qué es un problema de programación lineal?

- Un problema de programación lineal es un problema de programación matemática en el que la función objetivo es lineal y el conjunto factible viene definido por una colección finita de igualdades o desigualdades lineales.
- Todo problema de programación lineal admite una formulación del tipo:

$$\begin{array}{ll} max & c^t x \\ s.a: & Ax = b \\ & x \geq 0, \end{array}$$

donde A es una matriz $m \times n$, $b \in \mathbb{R}^m$, con $b \geq 0$, $c \in \mathbb{R}^n$ y $x \in \mathbb{R}^n$ es el vector variable cuyo valor debe ser determinado. Es lo que se llama la *forma estándar* del problema de programación lineal.

Un ejemplo

$$
\begin{array}{lll}
max & 5x_1 + 4x_2 & \\
s.a: & 6x_1 + 4x_2 & \leq 24 \\
 & x_1 + 2x_2 & \leq 6 \\
 & x_1 - x_2 & \geq 0 \\
 & x_1 & \geq 0 \\
 & x_2 & \geq 0
\end{array}
$$

Su forma estándar es:

$$
\begin{array}{ll}
max & 5x_1 + 4x_2 \\
s.a: & 6x_1 + 4x_2 + h_1 = 24 \\
 & x_1 + 2x_2 + h_2 = 6 \\
 & x_1 - x_2 - e_1 = 0 \\
 & x_1 \geq 0, x_2 \geq 0, h_1 \geq 0, h_2 \geq 0, e_1 \geq 0
\end{array}
$$

Un problema de producción

Consideremos una empresa de ingeniería que realiza dos tipos de proyectos, que denominaremos P_1 y P_2. En la empresa trabajan tres informáticos y un matemático (este último a tiempo parcial). La siguiente tabla contiene datos que indican la disponibilidad diaria de horas de trabajo de informáticos y matemáticos, el número de horas de trabajo de informáticos y matemáticos que son necesarias para completar un proyecto de cada tipo y los beneficios aproximados en miles de euros para cada tipo de proyecto. La política de la empresa impide que el número de proyectos P_2 que se realizan diariamente sea mayor que el de proyectos P_1. Se trata de determinar el número de proyectos de cada tipo que debe realizar la empresa diariamente para maximizar los beneficios.

	P_1	P_2	disponibilidad diaria
informáticos	6	4	24
matemáticos	1	2	6
beneficios	5	4	

Modelo asociado al problema de producción

1. Identificación de las variables.
   - x_1: número de proyectos P_1 realizados diariamente.
   - x_2: número de proyectos P_2 realizados diariamente.
2. Definir el objetivo: hay que maximizar el beneficio.

$$max\ 5x_1 + 4x_2$$

3. Definir las restricciones:
   - Restricción por la disponibilidad de informáticos: $6x_1 + 4x_2 \leq 24$
   - Restricción por la disponibilidad de informáticos: $x_1 + 2x_2 \leq 6$
   - Restricción por la política de la empresa: $x_1 - x_2 \geq 0$
   - Las variables son no negativas: $x_1 \geq 0$, $x_2 \geq 0$

Un ejemplo: el problema de producción

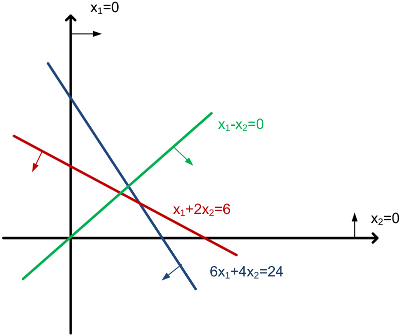
$$\begin{array}{lll} max & 5x_1 + 4x_2 & \\ s.a: & 6x_1 + 4x_2 & \leq 24 \\ & x_1 + 2x_2 & \leq 6 \\ & x_1 - x_2 & \geq 0 \\ & x_1 & \geq 0 \\ & x_2 & \geq 0 \end{array}$$

Su forma estándar es:

$$\begin{array}{ll} max & 5x_1 + 4x_2 \\ s.a: & 6x_1 + 4x_2 + h_1 = 24 \\ & x_1 + 2x_2 + h_2 = 6 \\ & x_1 - x_2 - e_1 = 0 \\ & x_1 \geq 0, x_2 \geq 0, h_1 \geq 0, h_2 \geq 0, e_1 \geq 0 \end{array}$$

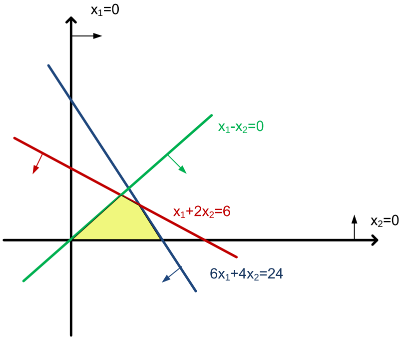
Resolución gráfica del problema de producción

$$
\begin{array}{lrl}
max & 5x_1 + 4x_2 & \\
s.a: & 6x_1 + 4x_2 & \leq 24 \\
& x_1 + 2x_2 & \leq 6 \\
& x_1 - x_2 & \geq 0 \\
& x_1 & \geq 0 \\
& x_2 & \geq 0
\end{array}
$$



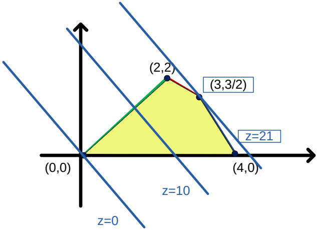
Resolución gráfica del problema de producción

$$
\begin{array}{lll}
max & 5x_1 + 4x_2 & \\
s.a: & 6x_1 + 4x_2 & \leq 24 \\
 & x_1 + 2x_2 & \leq 6 \\
 & x_1 - x_2 & \geq 0 \\
 & x_1 & \geq 0 \\
 & x_2 & \geq 0
\end{array}
$$



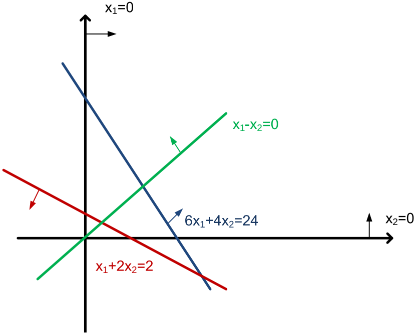
Resolución gráfica del problema de producción

$$\begin{array}{lll} max & 5x_1 + 4x_2 & \\ s.a: & 6x_1 + 4x_2 & \leq 24 \\ & x_1 + 2x_2 & \leq 6 \\ & x_1 - x_2 & \geq 0 \\ & x_1 & \geq 0 \\ & x_2 & \geq 0 \end{array}$$

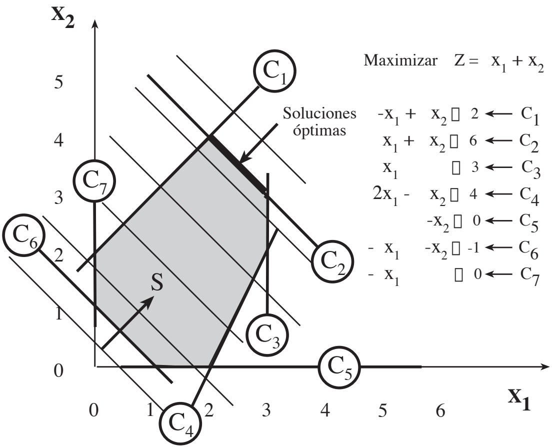


Un problema infactible

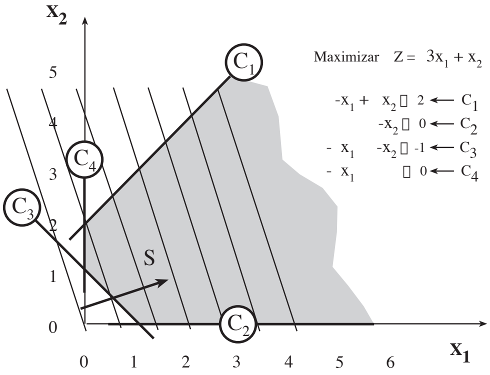
$$
\begin{array}{lll}
max & 5x_1 + 4x_2 & \\
s.a: & 6x_1 + 4x_2 & \geq 24 \\
 & x_1 + 2x_2 & \leq 2 \\
 & x_1 - x_2 & \leq 0 \\
 & x_1 & \geq 0 \\
 & x_2 & \geq 0
\end{array}
$$



Un problema con múltiples soluciones

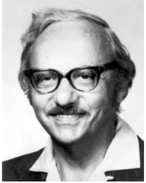


Un problema no acotado



Preliminares al algoritmo del símplex

- Es un algoritmo iterativo para resolver problemas de programación lineal en forma estándar.
- Fue formulado en 1947 por el matemático americano de origen ruso George Dantzig.
- Es probablemente la primera gran contribución de la investigación operativa.



George Dantzig (1914-2005). Nacido en Oregón en 1914, hijo de inmigrantes de origen ruso, estudia matemáticas en la Universidad de Maryland. Poco después de doctorarse por la Universidad de Berkeley, en 1947, formula el enunciado estándar de un problema general de Programación Lineal y desarrolla el método Simplex.

El paso a la forma estándar

Todo problema de programación lineal admite una presentación en forma estándar teniendo en cuenta las siguientes observaciones.

- Minimizar una función objetivo f es equivalente a maximizar $-f$.
- Las inecuaciones se convierten en ecuaciones introduciendo variables de holgura o de exceso.
- Si b tiene alguna componente negativa, basta cambiar de signo a ambos lados de la igualdad en la ecuación correspondiente.
- Si la variable x_i es libre ($x_i \in \mathbb{R}$) puede escribirse como $x_i = u_i - v_i$, con $u_i \geq 0$, $v_i \geq 0$.
- Si la variable x_i cumple que $x_i \leq 0$ puede hacerse el cambio $y_i = -x_i$, con $y_i \geq 0$.

Ejemplo del paso a la forma estándar

$$
\begin{aligned}
min \quad & 4x_1 + 3x_2 \\
s.a: \quad & x_1 + x_2 \leq 10 \\
& 2x_1 + x_2 \geq 2 \\
& 3x_1 + 4x_2 \leq -4 \\
& x_1 \leq 0
\end{aligned}
$$

Su forma estándar es:

$$
\begin{aligned}
max \quad & 4y_1 - 3u_2 + 3v_2 \\
s.a: \quad & -y_1 + u_2 - v_2 + h_1 = 10 \\
& -2y_1 + u_2 - v_2 - e_1 = 2 \\
& 3y_1 - 4u_2 + 4v_2 - e_2 = 4 \\
& y_1 \geq 0, u_2 \geq 0, v_2 \geq 0 \\
& h_1 \geq 0, e_1 \geq 0, e_2 \geq 0
\end{aligned}
$$

Fundamentos del algoritmo del símplex: forma estándar

Partimos de un problema de programación lineal en forma estándar.

$$\begin{array}{lll} max & c^t x & \\ s.a: & Ax = b & \quad (P1) \\ & x \geq 0, & \end{array}$$

donde A es una matriz $m \times n$, $b \in \mathbb{R}^m$, con $b \geq 0$, $c \in \mathbb{R}^n$ y $x \in \mathbb{R}^n$ es el vector variable. Normalmente nos encontraremos con problemas en los que:

- $m < n$, es decir, hay más variables que restricciones.
- Las m filas de la matriz A son linealmente independientes.

Para evitar trivialidades y dificultades no esenciales (y porque, en la práctica, no supondrá pérdida de generalidad) a partir de ahora trataremos con problemas que cumplen estas dos condiciones.

Fundamentos del algoritmo del símplex: soluciones básicas

- Sea $\bar{x}$ una solución de (P1) con $A\bar{x} = b$. Decimos que $\bar{x}$ es una *solución básica* de (P1) si todas las columnas de la matriz A asociadas a componentes de $\bar{x}$ distintas de cero son linealmente independientes. Una *solución básica factible* y una *solución básica óptima* son, respectivamente, una solución factible que además es básica y una solución óptima que además es básica.
- Se puede demostrar que si (P1) tiene una solución óptima (factible) entonces tiene una solución básica óptima (factible).
- Geométricamente, el conjunto factible de (P1) es un poliedro de $\mathbb{R}^m$ y una solución básica factible de (P1) es un punto extremo de su conjunto factible. Como un poliedro de $\mathbb{R}^m$ tiene un conjunto finito de puntos extremos, se puede buscar soluciones óptimas de (P1) explorando adecuadamente su conjunto finito de soluciones básicas factibles.

Fundamentos del algoritmo del símplex: forma canónica

Sea $\bar{x}$ una solución de (P1) con $A\bar{x} = b$.

- Si $\bar{x}$ es básica, podemos encontrar una submatriz B de A no singular y de dimensión $m \times m$ tal que contiene todas las columnas de A asociadas a componentes de $\bar{x}$ distintas de cero. En tal caso, decimos que $\bar{x}$ es una solución básica asociada a la base B.
- Obsérvese que el sistema $B^{-1}Ax = B^{-1}b$ es equivalente al sistema $Ax = b$ (en el sentido de que tienen las mismas soluciones) y cumple que todos los vectores de la base canónica de $\mathbb{R}^m$ son columnas de $B^{-1}A$. De un sistema cumpliendo tal condición diremos que está en forma canónica.
- En un problema de programación lineal en forma estándar cuyo sistema asociado está en forma canónica, es inmediato encontrar una solución básica factible.

Esquema general del algoritmo del símplex

1. Empezar obteniendo una solución básica factible o asegurarnos de que no tiene soluciones factibles (intentando poner en forma canónica el sistema de ecuaciones del problema de programación lineal en forma estándar).
2. Buscar una solución básica factible que mejore la que tenemos actualmente (en el sentido de que haga mayor la función objetivo).
3. Repetir el paso 2 hasta encontrar una solución óptima (una solución básica factible que no se pueda mejorar) o asegurarnos de que el problema no tiene solución óptima.

Un ejemplo

Sea el siguiente problema de programación lineal.

$$\begin{array}{rl} max & 5x_1 + 2x_2 + 3x_3 - x_4 + x_5 \\ s.a. & \quad x_1 + 2x_2 + 2x_3 + x_4 \qquad = 8 \\ & \quad 3x_1 + 4x_2 + x_3 + x_5 \qquad = 7 \\ & x_i \geq 0, \;\; i \in \{1, 2, 3, 4, 5\}. \end{array}$$

Obsérvese que el sistema asociado al problema está en forma canónica y, por lo tanto, la siguiente es una solución básica factible: $x_1 = x_2 = x_3 = 0$, $x_4 = 8$, $x_5 = 7$. El valor de la función objetivo asociado a esa solución es -1.

Beneficio relativo de una variable

El cambio neto en el valor la función objetivo para un incremento unitario de cada variable se denomina *beneficio relativo* y se denotará por $\bar{c}$.

Calculamos el beneficio relativo de x_1 (se mantiene $x_2 = x_3 = 0$). Si $x_1 = 1$ entonces:

$$x_4 = 7$$

$$x_5 = 4$$

$z = 2$ (es común denotar por z el valor de la función objetivo).

Luego aumentar el valor de x_1 en una unidad hace que el objetivo aumente en $2 - (-1) = 3$ unidades ($\bar{c}_1 = 3$). Sabemos, pues, que la anterior solución factible básica no es óptima. Haciendo el mismo razonamiento obtenemos $\bar{c}_2 = 0$ y $\bar{c}_3 = 4$.

Solución básica adyacente

Una solución básica *adyacente* a una dada difiere de esta última en una única variable básica.

En el ejemplo anterior, si incluimos en la base la variable que tenía el mejor beneficio relativo, x_3, el máximo valor que puede tomar ésta lo condicionan las restricciones:

$$2x_3 + x_4 = 8$$

$$x_3 + x_5 = 7$$

Como $x_4, x_5 \geq 0$ se tiene

$$x_4 = 8 - 2x_3 \geq 0 \Rightarrow x_3 \leq 4$$

$$x_5 = 7 - x_3 \geq 0 \Rightarrow x_3 \leq 7$$

Por tanto, $x_3 \leq 4$ y la variable x_4 se anula, pasando a no ser básica. La solución así obtenida es adyacente a la original.

Esquema detallado del algoritmo del símplex (I)

- Paso 1. Empezar obteniendo una solución básica factible poniendo el sistema en forma canónica (si ello es posible).
- Paso 2. Comprobar si esa solución es óptima, calculando los beneficios relativos de las variables no básicas. Si todos son ≤ 0, la solución actual es óptima y se finaliza. En otro caso, seguir.
- Paso 3. Seleccionar la variable no básica con el mayor beneficio relativo (en caso de empate elegimos una cualquiera). Esta variable entrará en la base.

Esquema detallado del algoritmo del símplex (II)

- Paso 4. Elegir la variable que sale de la base por la regla de la mínima proporción: las restricciones en las que la variable no básica seleccionada tiene coeficiente positivo restringen el crecimiento de tal variable al cociente entre la constante de la derecha y el valor del coeficiente positivo; sale de la base la variable básica que se corresponde con la fila en la que se alcanza el mínimo de tales cocientes (en caso de empate elegimos una cualquiera). Los ceros o negativos no limitan el crecimiento de la variable que entra; si todas las restricciones tienen coeficiente menor o igual a cero, el problema es no acotado.
- Paso 5. Obtener un nuevo sistema equivalente al original en el que la matriz asociada a la nueva solución básica factible sea la identidad. Volver al paso 2.

Un ejemplo

Resolvamos el siguiente problema usando el método símplex.

$$
\begin{array}{lll}
max & 5x_1 + 2x_2 + 3x_3 - x_4 + x_5 & \\
s.a: & x_1 + 2x_2 + 2x_3 + x_4 & = 8 \\
 & 3x_1 + 4x_2 + x_3 + x_5 & = 7 \\
 & x_i \geq 0 &
\end{array}
$$

Coeficientes del objetivo

b

Variables básicas

A

	5	2	3	-1	1	
-1	1	2	2	1	0	8
1	3	4	1	0	1	7
	3	0	4	0	0	-1

Beneficios relativos

z

Un ejemplo

- Para calcular los beneficios relativos:

$$\bar{c}_i = c_i - c_B P_i$$

  donde c_B son los coeficientes en el objetivo de las variables básicas y P_i es la columna en la tabla de la variable x_i.
- Tras un pivotaje:
  - Nueva fila pivote = fila antigua pivote divida por el valor del pivote.
  - Nueva fila = fila antigua-(coeficiente de la columna pivote)$\times$(nueva fila pivote)

Un ejemplo

TABLA 1

	5	2	3	-1	1	
-1	1	2	2	1	0	8
1	3	4	1	0	1	7
	3	0	4	0	0	-1

TABLA 2

	5	2	3	-1	1	
3	1/2	1	1	1/2	0	4
1	5/2	3	0	-1/2	1	3
	1	-4	0	-2	0	15

TABLA 3

	5	2	3	-1	1	
3	0	2/5	1	3/5	-1/5	17/5
			Óptima			
5	1	6/5	0	-1/5	2/5	6/5
	0	-26/5	0	-9/5	-2/5	81/5

Comentarios al método símplex

- Si la columna de la variable seleccionada para entrar en la base en una iteración tiene todos sus elementos menores o iguales que cero, entonces el problema es no acotado.
- Si hay empates en la regla de la mínima proporción alguna variable básica tomará el valor cero. A ese tipo de soluciones se les llama *degeneradas*. La presencia de soluciones degeneradas implicará normalmente un aumento en el número de iteraciones.
- Si en la tabla óptima alguna variable no básica tiene beneficio relativo igual a cero, entonces existen infinitas soluciones óptimas.
- A veces no es fácil obtener una solución básica factible para iniciar el algoritmo. De hecho, puede no existir tal solución (cuando el problema es infactible). Para tratar estas cuestiones se utilizan *variables artificiales*.

Obtención de una solución básica factible inicial

- Supongamos que tenemos el siguiente problema:

$$\begin{aligned} x_1 - 2x_2 + x_3 &= 2 \\ x_1 - x_2 - 4x_3 &= 4. \end{aligned}$$

- Se puede poner

$$\begin{aligned} x_1 - 2x_2 + x_3 + x_4 &= 2 \\ x_1 - x_2 - 4x_3 + x_5 &= 4 \end{aligned}$$

donde x_4 y x_5 son variables artificiales. Cualquier solución factible de este nuevo problema en la que las variables artificiales valgan cero da lugar a una solución factible del problema original.

Obtención de una solución básica factible inicial

Con el objeto de anular las variables artificiales se suele utilizar alguno de los dos métodos siguientes.

1. **El método de penalizaciones.** Consiste en incluir las variables artificiales en la función objetivo penalizándolas con un coeficiente muy pequeño $-M$. Si en el óptimo alguna variable artificial toma un valor positivo, entonces el problema original es infactible.
2. **El método de las dos fases.** Resuelve un primer problema que minimiza la suma de las variables artificiales en el conjunto factible del problema original. Si se consigue que éstas valgan cero, se pasa a la siguiente fase cambiando la función objetivo por la original y continuando las iteraciones. Si no se consigue que todas las variable artificiales valgan cero, entonces el problema original es infactible.

Comentarios computacionales al método símplex

- En la práctica, el método símplex es computacionalmente eficiente y se puede utilizar en general para problemas de gran tamaño (con miles de restricciones y cientos de miles de variables).
- Sin embargo, aunque demuestra un funcionamiento muy satisfactorio en la solución de la mayoría de los problemas prácticos, no es un algoritmo en tiempo polinomial, sino en tiempo exponencial. Esto significa que, en el caso más desfavorable, su tiempo de computación puede crecer exponencialmente con el tamaño del problema.
- En 1984 N.K. Karmarkar introdujo un nuevo algoritmo basado en un método de punto interior para resolver problemas de programación lineal. Además demostró que su algoritmo funciona en tiempo polinomial. En la práctica, es competitivo con el método símplex. De hecho, actualmente muchos *solvers* dan la posibilidad de aplicar el símplex o el método de punto interior.

El problema

En la siguiente tabla se muestra el conjunto de actividades que componen un proyecto, sus duraciones en días y las relaciones de precedencia inmediata entre ellas. Se trata de plantear un problema de programación lineal cuya solución nos dé la duración mínima del proyecto.

Actividad	1	2	3	4	5	6	7	8	9	10
Precedentes Inmediatas	-	-	2	3	1	-	1	4,6,7	5	9
Duración	2	3	2	4	5	2	1	3	2	2

El modelo

- $x_i =$ instante de comienzo de la actividad i ($i \in \{1, \ldots, 10\}$).
- $x_{11} =$ instante de finalización del proyecto.

$$
\begin{array}{ll}
min & x_{11} \\
s.a: & x_2 + 3 \leq x_3 \\
 & x_3 + 2 \leq x_4 \\
 & x_1 + 2 \leq x_5 \\
 & x_1 + 2 \leq x_7 \\
 & x_4 + 4 \leq x_8 \\
 & x_6 + 2 \leq x_8 \\
 & x_7 + 1 \leq x_8 \\
 & x_5 + 5 \leq x_9 \\
 & x_9 + 2 \leq x_{10} \\
 & x_8 + 3 \leq x_{11} \\
 & x_{10} + 2 \leq x_{11} \\
 & x_i \geq 0, i \in \{1, \ldots, 11\}
\end{array}
$$

Código en R

```
install.packages("lpSolve")
library(lpSolve)
cv<-c(rep(0,10),1)
bv<-c(-3,-2,-2,-2,-4,-2,-1,-5,-2,-3,-2)
restmat<-matrix(c(0,1,-1, rep(0,8),
0,0,1,-1, rep(0,7),
1,0,0,0,-1, rep(0,6),
1,rep(0,5),-1, rep(0,4),
0,0,0,1,0,0,0,-1, 0,0,0,
rep(0,5),1,0,-1,0,0,0,
rep(0,6),1,-1,0,0,0,
rep(0,4),1,0,0,0,-1,0,0,
rep(0,8),1,-1,0,
rep(0,7),1,0,0,-1,
rep(0,9),1,-1),nrow=11,byrow=TRUE)
dirv<-c(rep(«=",11))
res<-lp("min",cv,restmat,dirv,bv)
res
sol<-lp("min",cv,restmat,dirv,bv)$solution
sol
```

Bibliografía

1. Bazaraa, M.S. (2005). Programación Lineal y Flujo en Redes. Limusa.
2. Cornuejols, G. and Tütüncü, R. (2011) Optimization Methods in Finance. Cambridge University Press.
3. Bazaraa, M.S., Jarvis, J.J. and Sherali, H.D. (2010) Linear Programming and Network Flows. Wiley.