Simulación y Optimización

Programación lineal
Parte II

Enrique E. Tarifa, Facultad de Ingeniería, UNJu

Programación matemática

Programación matemática

$$\underset{x_j}{\text{Max}}\, FO\left(x_1, x_2, \ldots, x_n\right) \quad j = 1, 2, \ldots, n$$

s. a:

$$h_i\left(x_1, x_2, \ldots, x_n\right) = 0 \quad i = 1, 2, \ldots, m$$

$$g_k\left(x_1, x_2, \ldots, x_n\right) \leq 0 \quad k = 1, 2, \ldots, o$$

$$GL = n - m$$

Clasificación de modelos

- Programación Lineal (LP, *Linear Programming*)
- Programación No Lineal (NLP, *Nonlinear Programning*)
- Programación Lineal Entera (ILP, *Integer Linear Programming*)
- Programación No Lineal Entera (INLP, *Integer Nonlinear Programming*)
- Programación Lineal Entera Mixta (MILP, *Mixed Integer Linear Programming*)
- Programación No Lineal Entera Mixta (MINLP, *Mixed Integer Nonlinear Programming*)

$$FO(x_1, x_2, \ldots, x_n) = a_1 x_1 + a_2 x_2 + \ldots + a_n x_n$$

Programación Lineal

Programación lineal

$$\underset{x_j}{\text{Max}} \quad c_1\,x_1 + c_2\,x_2 + \ldots + c_j\,x_j + \ldots + c_n\,x_n \qquad j = 1, 2, \ldots, n$$

s. a:

$$a_{11}\,x_1 + a_{12}\,x_2 + \ldots + a_{1j}\,x_j + \ldots + a_{1n}\,x_n \le b_1$$

$$a_{21}\,x_1 + a_{22}\,x_2 + \ldots + a_{2j}\,x_j + \ldots + a_{2n}\,x_n \le b_2$$

$$\ldots$$

$$a_{i1}\,x_1 + a_{i2}\,x_2 + \ldots + a_{ij}\,x_j + \ldots + a_{in}\,x_{in} \le b_i$$

$$\ldots$$

$$a_{m1}\,x_1 + a_{m2}\,x_2 + \ldots + a_{mj}\,x_j + \ldots + a_{mn}\,x_{mn} \le b_m$$

$$x_j \ge 0 \quad j = 1 \ldots n, b_i \ge 0 \quad i = 1 \ldots m$$

$$GL = n$$

Programación lineal

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{pmatrix} \qquad c = \begin{pmatrix} c_1 \\ c_2 \\ \cdots \\ c_n \end{pmatrix} \qquad b = \begin{pmatrix} b_1 \\ b_2 \\ \cdots \\ b_m \end{pmatrix} \qquad A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

Programación lineal

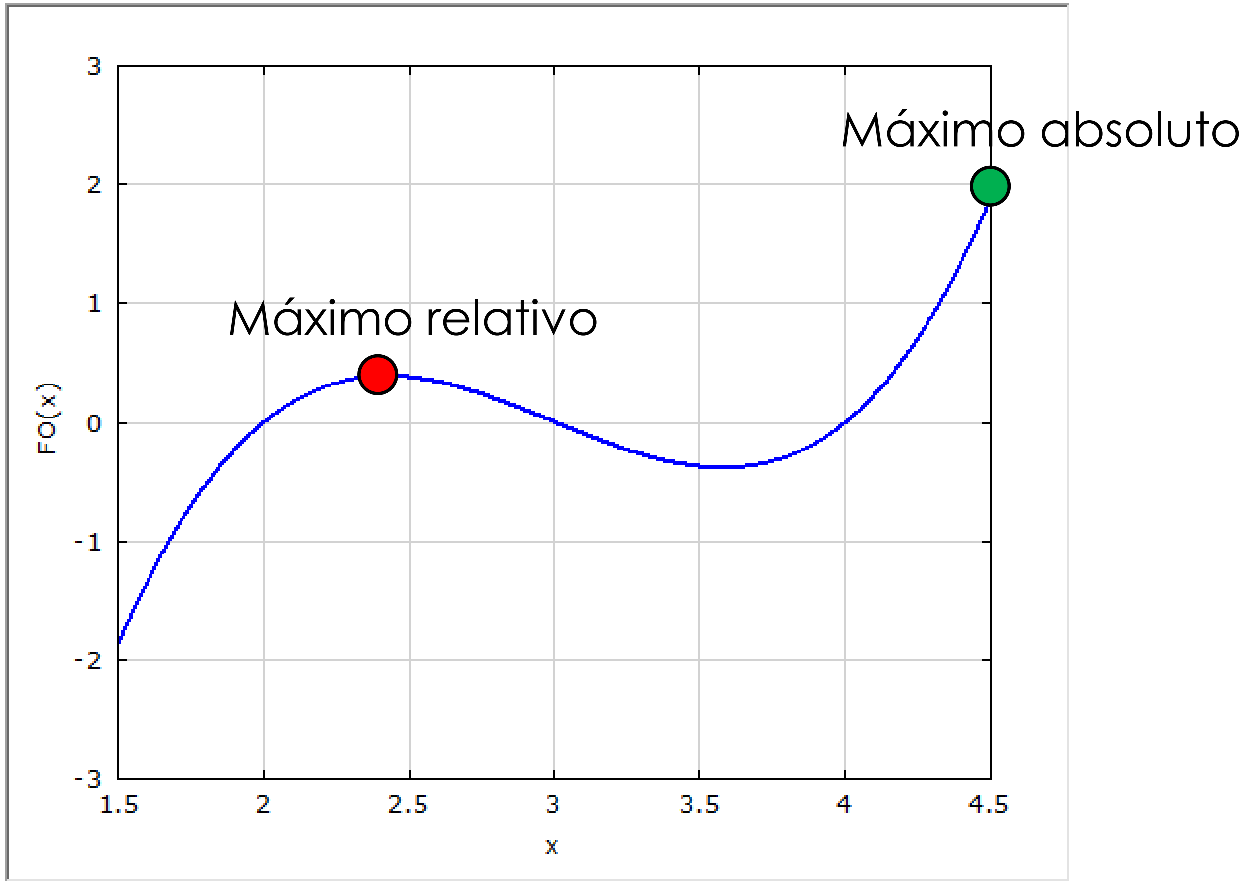
$$\underset{x}{\text{Max}}\, c^{\mathrm{T}} x$$

s. a:

$$Ax \leq b$$

$$x \geq 0 \quad b \geq 0$$

Modelo no lineal



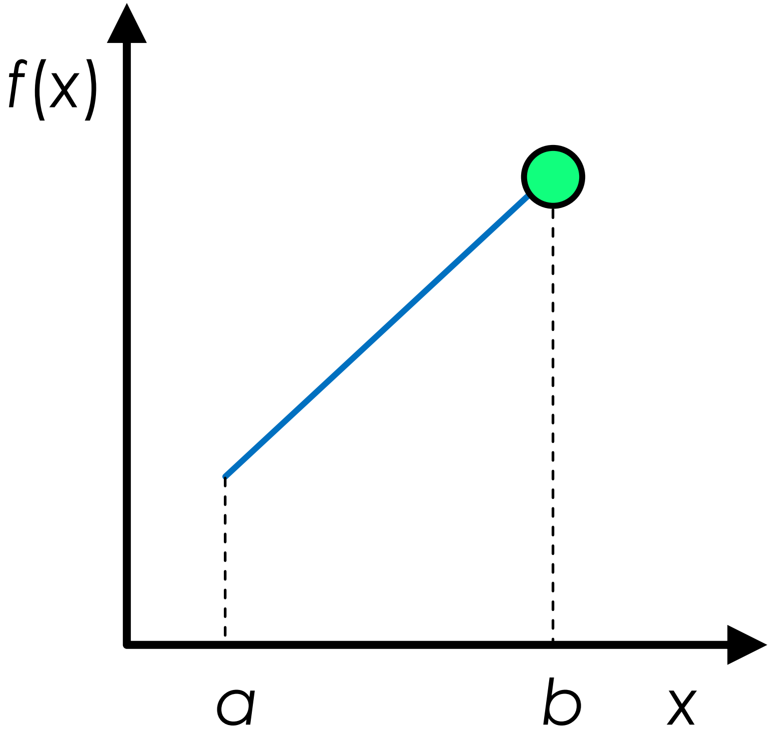
Programación lineal

$$\underset{x}{\text{Max}}\ f\left(x\right)$$

s. a:

$$x \geq a$$

$$x \leq b$$



- El óptimo es global.
- Es único.
- Está en un extremo.

Problema de mezcla

Problema de mezcla

Se desea producir un producto mezclando s ingredientes considerando t propiedades de la mezcla. La cantidad a utilizar de cada ingrediente j es n_j (kg). El costo unitario del ingrediente j es c_j ($/kg), y la cantidad disponible es n_j^0. A su vez, el ingrediente j tiene un contenido específico de propiedad k igual a w_{jk}. El contenido específico w_k^f de la propiedad k en el producto debe respetar las especificaciones de contenido mínimo ($\alpha_k = 1$) o máximo ($\alpha_k = -1$) w_k^0. El problema consiste en determinar los valores de n_j que minimicen el costo.

Problema de mezcla

- Parámetros:
  - s: Tipos de ingredientes.
  - t: Tipos de propiedades.
  - m: Cantidad del producto (kg).
  - c_j: Costo unitario del ingrediente j (\$/kg).
  - n_j^0 : Cantidad disponible del ingrediente j (kg).
  - w_{jk}: Contenido específico en el ingrediente j de la propiedad k.
  - α_k: Factor que indica si la propiedad k es deseable ($\alpha_k = 1$) o no ($\alpha_k = -1$) en el producto.
  - w_k^0: Cota mínima o máxima de la propiedad k en el producto.

Problema de mezcla

- Variables de decisión:
  - n_j: Cantidad del ingrediente j en el producto (kg).
  - w_k^{f}: Contenido específico en el producto de la propiedad k.

Problema de mezcla

- Función objetivo:
  - Costo ($): $\sum_{j=1}^{s} c_j n_j$
- Restricciones:
  - Balance de materia: $\sum_{j=1}^{s} n_j = m$
  - Inventario: $n_j \leq n_j^0 \quad \forall j$
  - Balance de propiedad: $\sum_{j=1}^{s} w_{jk} n_j = w_k^{\mathrm{f}} m \quad \forall k$
  - Requerimientos: $\alpha_k w_k^{\mathrm{f}} \geq \alpha_k w_k^0 \quad \forall k$

Problema de mezcla

$$\underset{n_j, w_k^{\mathrm{f}}}{\mathrm{Min}} \sum_{j=1}^{s} c_j\, n_j$$

s. a:

$$\sum_{j=1}^{s} n_j = m$$

$$n_j \le n_j^0 \qquad \forall j$$

$$\sum_{j=1}^{s} w_{jk}\, n_j = w_k^{\mathrm{f}}\, m \qquad \forall k$$

$$\alpha_k\, w_k^{\mathrm{f}} \ge \alpha_k\, w_k^0 \qquad \forall k$$

$$GL = s + t - (1 + t) = s - 1$$

Problema de mezcla con composición

Problema de mezcla con composición

Se desea producir un producto mezclando s ingredientes considerando t propiedades de la mezcla. La fracción peso de cada ingrediente j es x_j. El costo unitario del ingrediente j es c_j ($/kg). A su vez, el ingrediente j tiene un contenido específico de propiedad k igual a w_{jk}. El contenido específico w_k^{f} de la propiedad k en el producto debe respetar las especificaciones de contenido mínimo $(\alpha_k = 1)$ o máximo $(\alpha_k = -1)$ w_k^0. El problema consiste en determinar los valores de x_j que minimicen el costo.

Problema de mezcla con composición

- Parámetros:
  - s: Tipos de ingredientes.
  - t: Tipos de propiedades.
  - c_j: Costo unitario del ingrediente j (\$/kg).
  - w_{jk}: Contenido específico en el ingrediente j de la propiedad k.
  - α_k: Factor que indica si la propiedad k es deseable ($\alpha_k = 1$) o no ($\alpha_k = -1$) en el producto.
  - w_k^0: Cota mínima o máxima de la propiedad k en el producto.

Problema de mezcla con composición

- Variables de decisión:
  - x_j: Fracción peso del ingrediente j en el producto.
  - w_k^{f}: Contenido específico en el producto de la propiedad k.

Problema de mezcla con composición

- Función objetivo:
  - Costo ($/kg): $\sum_{j=1}^{s} c_j x_j$
- Restricciones:
  - Balance de materia: $\sum_{j=1}^{s} x_j = 1$
  - Balance de propiedad: $\sum_{j=1}^{s} w_{jk} x_j = w_k^{\mathrm{f}} \quad \forall k$
  - Requerimientos: $\alpha_k w_k^{\mathrm{f}} \geq \alpha_k w_k^{0} \quad \forall k$

Problema de mezcla con composición

$$\underset{x_j, w_k^{\mathrm{f}}}{\mathrm{Min}} \sum_{j=1}^{s} c_j \, x_j$$

s. a:

$$\sum_{j=1}^{s} x_j = 1$$

$$\sum_{j=1}^{s} w_{jk} \, x_j = w_k^{\mathrm{f}} \qquad \forall k$$

$$\alpha_k \, w_k^{\mathrm{f}} \geq \alpha_k \, w_k^{0} \qquad \forall k$$

$$GL = s + t - (1 + t) = s - 1$$

Alimento balanceado

Alimento balanceado

- En una planta productora de alimento balanceado, la formulación de un tipo de alimento requiere dos ingredientes activos y un soporte para dar volumen.
- Contenido específico mínimo de nutrientes en el producto (g/kg):

Nutriente	A	B	C	D
Cantidad	90	50	20	2

Alimento balanceado

- Contenido específico de nutrientes (g/kg) y costo ($/kg) de los ingredientes:

	A	B	C	D	Costo
Ingrediente 1	100	80	40	10	40
Ingrediente 2	200	150	20	0	60
Soporte	0	0	0	0	0

Alimento balanceado

- Parámetros:
  - $s = 3$
  - $t = 4$
  - $\alpha_k = 1$, todas las propiedades son deseables.

Alimento balanceado

- Variables de decisión:
  - x_1: Fracción peso del ingrediente 1 en el producto.
  - x_2: Fracción peso del ingrediente 2 en el producto.
  - x_3: Fracción peso del soporte en el producto.

Alimento balanceado

- Función objetivo:
  - Costo ($/kg): $40x_1 + 60x_2$
- Restricciones:
  - Balance de materia: $x_1 + x_2 + x_3 = 1$
  - Cantidad de nutriente A (g/kg): $100x_1 + 200x_2 \geq 90$
  - Cantidad de nutriente B (g/kg): $80x_1 + 150x_2 \geq 50$
  - Cantidad de nutriente C (g/kg): $40x_1 + 20x_2 \geq 20$
  - Cantidad de nutriente D (g/kg): $10x_1 \geq 2$

Alimento balanceado

$$\min_{x_j, w_k^{\mathrm{f}}} \sum_{j=1}^{s} c_j \, x_j$$

s. a:

$$\sum_{j=1}^{s} x_j = 1$$

$$\sum_{j=1}^{s} w_{jk} \, x_j = w_k^{\mathrm{f}} \quad \forall k$$

$$\alpha_k \, w_k^{\mathrm{f}} \geq \alpha_k \, w_k^{0} \quad \forall k$$

$$\min_{x_1, x_2, x_3} 40\, x_1 + 60\, x_2$$

s.a:

$$x_1 + x_2 + x_3 = 1$$

$$100\, x_1 + 200\, x_2 \geq 90$$

$$80\, x_1 + 150\, x_2 \geq 50$$

$$40\, x_1 + 20\, x_2 \geq 20$$

$$10\, x_1 \geq 2$$

x_1, x_2, x_3 no negativas

Alimento balanceado.lg4

Modelo en LINGO

Lingo Model - Alimento balanceado

```
! Alimento balanceado;
[FO] Min = 40*x1+60*x2;

[RM] x1+x2+x3 = 1;
[RA] 100*x1+200*x2 >= 90;
[RB] 80*x1+150*x2 >= 50;
[RC] 40*x1+20*x2 >= 20;
[RD] 10*x1 >= 2;
```

Resultados en LINGO

Variable	Value	Reduced Cost
X1	0.3666667	0.000000
X2	0.2666667	0.000000
X3	0.3666667	0.000000

Row	Slack or Surplus	Dual Price
FO	30.66667	-1.000000
RM	0.000000	0.000000
RA	0.000000	-0.2666667
RB	19.33333	0.000000
RC	0.000000	-0.3333333
RD	1.666667	0.000000

Problema de transporte

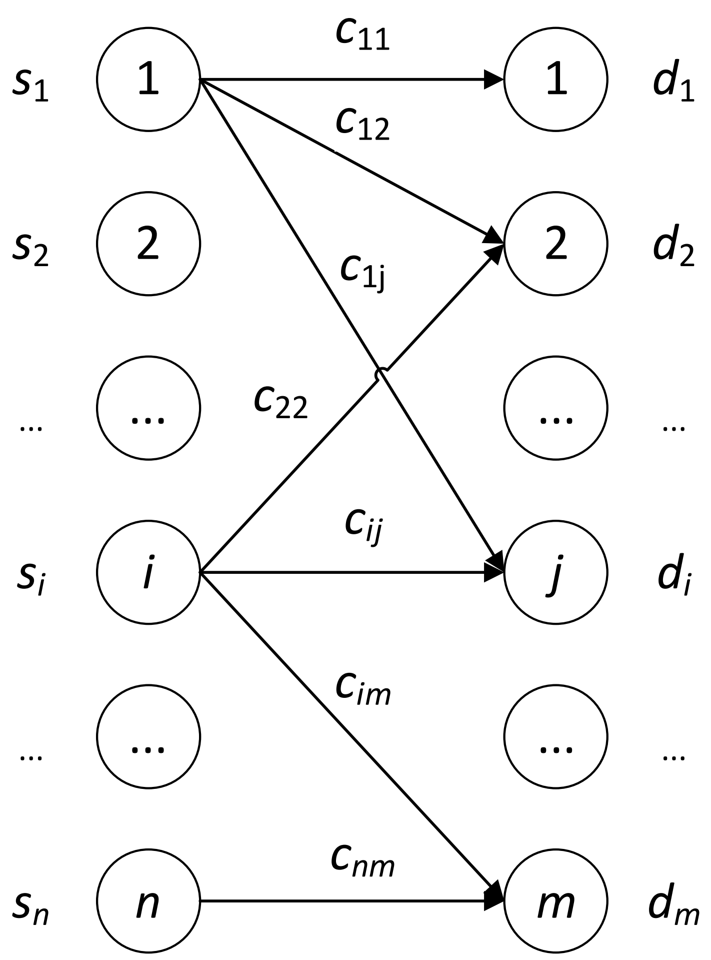
Problema del transporte

Se debe distribuir un bien desde *m* centros de abastecimiento, llamados orígenes, a *n* centros de recepción, llamados destinos. El origen *i* dispone de s_i unidades para distribuir, y el destino *j* tiene una demanda de d_j unidades que debe ser satisfecha por los orígenes. El costo para enviar una unidad desde el origen *i* al destino *j* es c_{ij}. Se desea determinar la cantidad de unidades x_{ij} que se deben enviar desde cada origen *i* a cada destino *j* para que el costo de transporte sea mínimo.

Problema del transporte

- Parámetros:
  - m: Cantidad de orígenes.
  - n: Cantidad de destinos.
  - s_i: Disponibilidad en el origen i.
  - d_j: Demanda en el destino j.
  - c_{ij}: Costo unitario de transporte de i a j.
- Variables de decisión:
  - x_{ij}: Cantidad enviada de i a j.

Problema del transporte



X_{ij}

Problema de transporte

- Casos:
  - Oferta igual a la demanda: $\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j$
  - Oferta mayor que la demanda: $\sum_{i=1}^{m} s_i > \sum_{j=1}^{n} d_j$
  - Oferta menor que la demanda: $\sum_{i=1}^{m} s_i < \sum_{j=1}^{n} d_j$

Problema de transporte

- Oferta igual a la demanda: $\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j$
- Función objetivo:
  - Costo (\$) = $\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$
- Restricciones:
  - Origen *i* (caja): $\sum_{j=1}^{n} x_{ij} = s_i$
  - Destino *j* (caja): $\sum_{i=1}^{m} x_{ij} = d_j$

Problema del transporte

$$\operatorname*{Min}_{x_{ij}} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

s. a:

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

$$\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j$$

$$\sum_{i=1}^{m} x_{ij} = d_j \quad \forall j$$

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

Problema del transporte

$$\underset{x_{ij}}{\text{Min}} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

s. a:

$$\sum_{j=1}^{n} x_{ij} \leq s_i \quad \forall i$$

$$\sum_{i=1}^{m} s_i > \sum_{j=1}^{n} d_j$$

$$\sum_{i=1}^{m} x_{ij} = d_j \quad \forall j$$

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

Problema del transporte

$$\operatorname*{Min}_{x_{ij}} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

s. a:

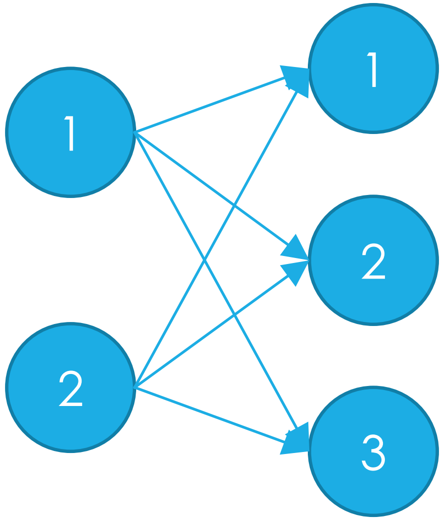
$$\sum_{j=1}^{n} x_{ij} = s_i \quad \forall i$$

$$\sum_{i=1}^{m} s_i < \sum_{j=1}^{n} d_j$$

$$\sum_{i=1}^{m} x_{ij} \leq d_j \quad \forall j$$

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

Ejemplo de 2x3



$$\underset{x_{ij}}{\text{Min}} \sum_{i=1}^{2}\sum_{j=1}^{3} c_{ij}\,x_{ij} = c_{1,1}x_{1,1} + c_{1,2}x_{1,2} + c_{1,3}x_{1,3}$$

$$+c_{2,1}x_{2,1} + c_{2,2}x_{2,2} + c_{2,3}x_{2,3}$$

s. a:

$$x_{1,1} + x_{1,2} + x_{1,3} = s_1$$

$$x_{2,1} + x_{2,2} + x_{2,3} = s_2$$

$$x_{1,1} + x_{2,1} = d_1$$

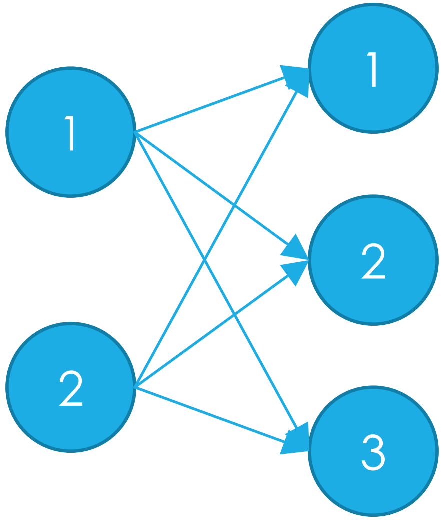
$$x_{1,2} + x_{2,2} = d_2$$

$$x_{1,3} + x_{2,3} = d_3$$

Total unimodularidad

- Una matriz es totalmente unimodular si todas sus submatrices cuadradas tienen determinante 0, 1 o -1. Esta propiedad garantiza que, cuando se tiene un problema de programación lineal (LP) con una matriz de restricciones que es totalmente unimodular y un vector de términos independientes enteros, cualquier vértice (solución básica factible) del poliedro de soluciones será entero.
- Si los s_i y los d_j son enteros, $x_{ij} \in \mathbb{N}\ \forall i, \forall j$.

Ejemplo de 2x3



$$\underset{x_{ij}}{\text{Min}} \sum_{i=1}^{2}\sum_{j=1}^{3} c_{ij}\,x_{ij} = c_{1,1}\,x_{1,1} + c_{1,2}\,x_{1,2} + c_{1,3}\,x_{1,3}$$

$$+c_{2,1}\,x_{2,1} + c_{2,2}\,x_{2,2} + c_{2,3}\,x_{2,3}$$

s. a:

$$x_{1,1} + x_{1,2} + x_{1,3} = s_1$$

$$x_{2,1} + x_{2,2} + x_{2,3} = s_2$$

$$x_{1,1} + x_{2,1} = d_1$$

$$x_{1,2} + x_{2,2} = d_2$$

$$x_{1,3} + x_{2,3} = d_3$$

Distribuidora de insumos

Distribuidora de insumos

Una distribuidora de insumos debe abastecer a 8 fábricas. Para ello, cuenta con 6 depósitos estratégicamente ubicados.

Distribuidora de insumos

Depósito	Disponibilidad (caja)
DE1	60
DE2	55
DE3	51
DE4	43
DE5	41
DE6	52
Total	**302**

Fábricas	Demanda (caja)
FA1	35
FA2	37
FA3	22
FA4	32
FA5	41
FA6	32
FA7	43
FA8	38
Total	**280**

Distribuidora de insumos

Costo de transporte ($/caja)

	FA1	FA2	FA3	FA4	FA5	FA6	FA7	FA8
DE1	6	2	6	7	4	2	5	9
DE2	4	9	5	3	8	5	8	2
DE3	5	2	1	9	7	4	3	3
DE4	7	6	7	3	9	2	7	1
DE5	2	3	9	5	7	2	6	5
DE6	5	5	2	2	8	1	4	3

Distribuidora de insumos.lg4

Distribuidora de insumos

$$\underset{x_{ij}}{\text{Min}} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\, x_{ij}$$

s. a:

$$\sum_{j=1}^{n} x_{ij} \leq s_i \quad \forall i$$

$$\sum_{i=1}^{m} s_i > \sum_{j=1}^{n} d_j$$

$$\sum_{i=1}^{m} x_{ij} = d_j \quad \forall j$$

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

```
!Distribuidora de insumos;
SETS:
    DEPOSITOS/DE1..DE6/: S;
    FABRICAS/FA1..FA8/: D;
    DXF(DEPOSITOS,FABRICAS): C,X;
ENDSETS

! Los datos;
DATA:
    S = 60 55 51 43 41 52;
    D = 35 37 22 32 41 32 43 38;
    C = 6 2 6 7 4 2 5 9
        4 9 5 3 8 5 8 2
        5 2 1 9 7 4 3 3
        7 6 7 3 9 2 7 1
        2 3 9 5 7 2 6 5
        5 5 2 2 8 1 4 3;
ENDDATA

! Función objetivo;
[FO] MIN = @SUM(DXF: C*X);

! Restricciones de orígenes;
@FOR(DEPOSITOS(I):
[RO] @SUM(FABRICAS(J): X(I,J)) <= S(I)
);

! Restricciones de destinos;
@FOR(FABRICAS(J):
[RD] @SUM(DEPOSITOS(I): X(I,J)) = D(J)
);
```

Modelo en LINGO

$$\operatorname*{Min}_{x_{ij}} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\, x_{ij}$$

s. a:

$$\sum_{j=1}^{n} x_{ij} \leq s_i \quad \forall i$$

$$\sum_{i=1}^{m} x_{ij} = d_j \quad \forall j$$

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

$$\sum_{i=1}^{m} s_i > \sum_{j=1}^{n} d_j$$

Lingo Model - Distribuidora de insumos

```
!Distribuidora de insumos;
SETS:
   DEPOSITOS/DE1..DE6/: S;
   FABRICAS/FA1..FA8/: D;
   DXF(DEPOSITOS,FABRICAS): C,X;
ENDSETS

! Los datos;
DATA:
   S = 60 55 51 43 41 52;
   D = 35 37 22 32 41 32 43 38;
   C = 6 2 6 7 4 2 5 9
       4 9 5 3 8 5 8 2
       5 2 1 9 7 4 3 3
       7 6 7 3 9 2 7 1
       2 3 9 5 7 2 6 5
       5 5 2 2 8 1 4 3;
ENDDATA

! Función objetivo;
[FO] MIN = @SUM(DXF: C*X);

! Restricciones de orígenes;
@FOR(DEPOSITOS(I):
[RO] @SUM(FABRICAS(J): X(I,J)) <= S(I)
);

! Restricciones de destinos;
@FOR(FABRICAS(J):
[RD] @SUM(DEPOSITOS(I): X(I,J)) = D(J)
);
```

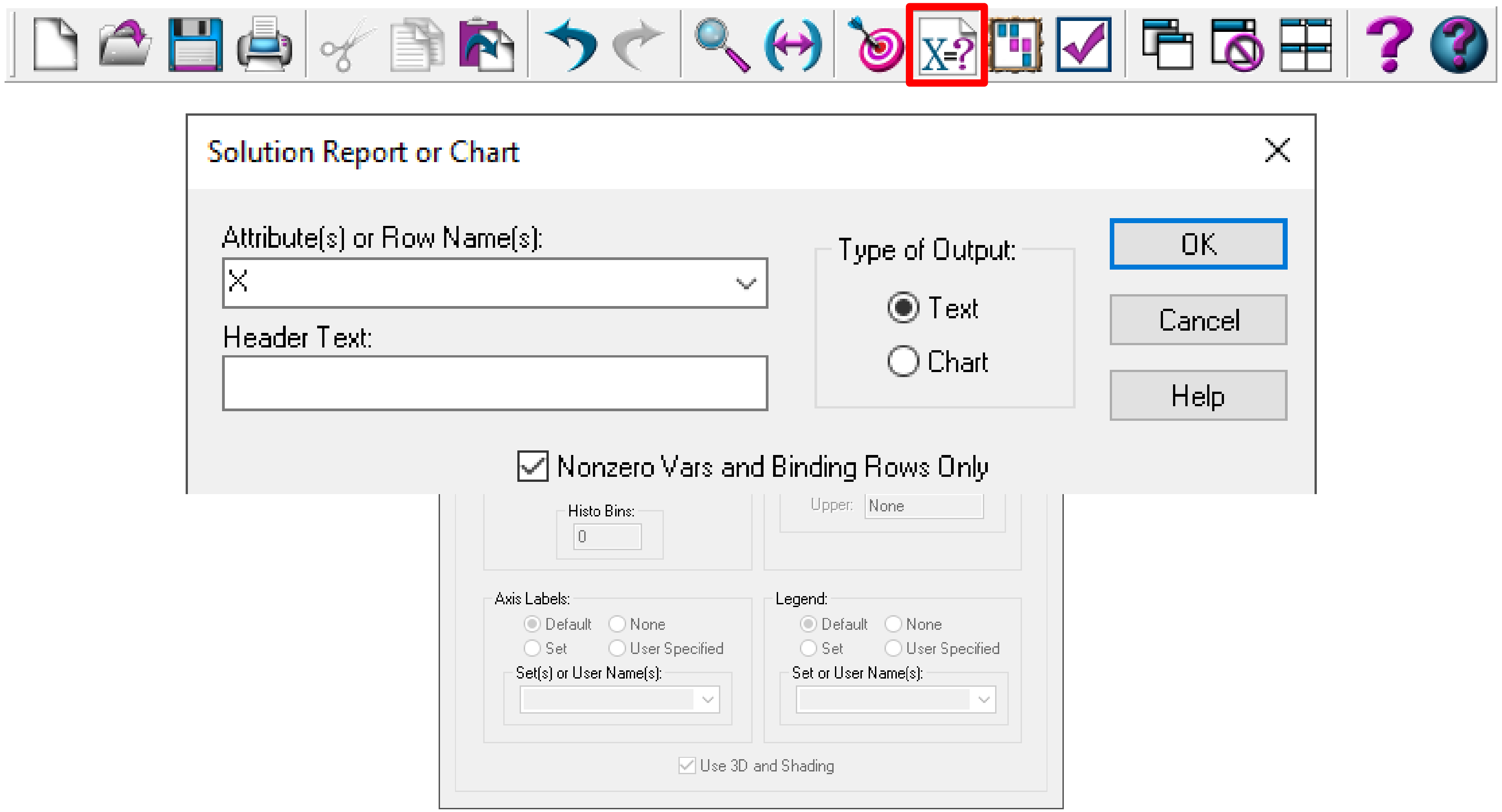
Resultados en LINGO

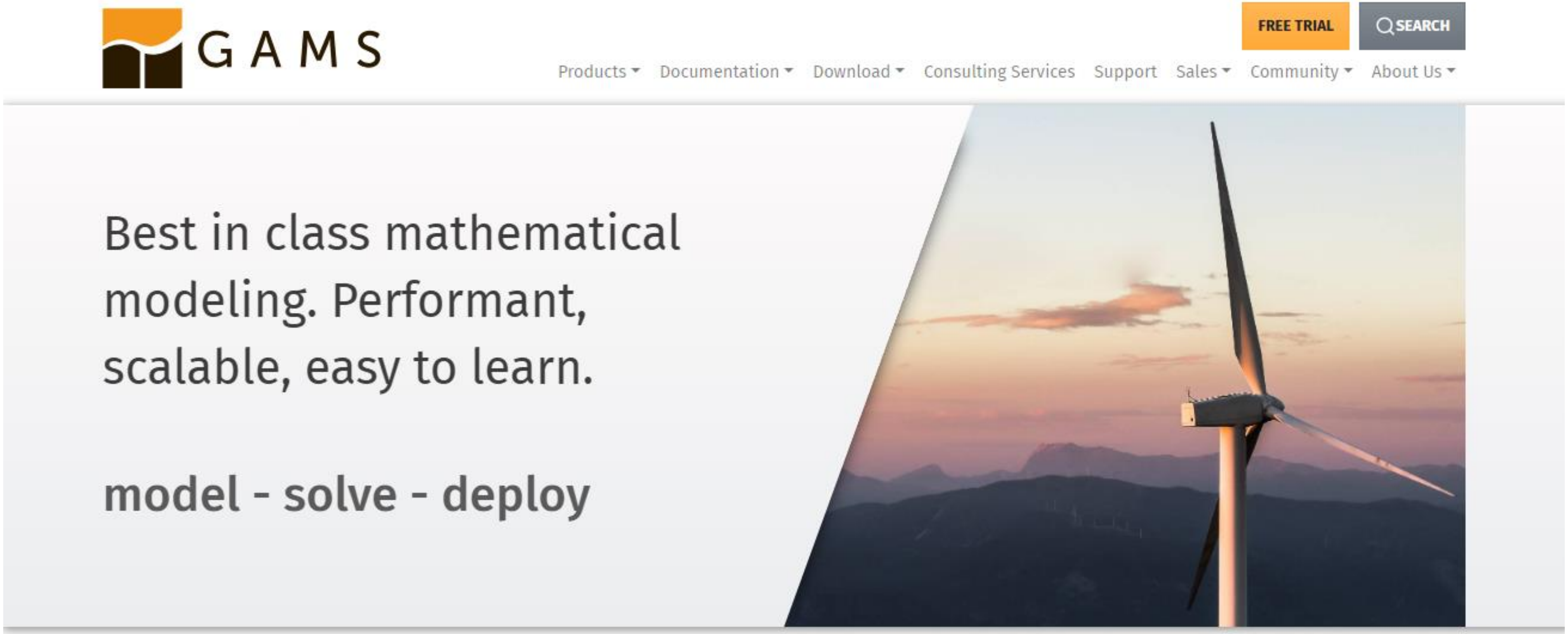
Variable	Value	Reduced Cost
X(DE1, FA2)	19.00000	0.000000
X(DE1, FA5)	41.00000	0.000000
X(DE2, FA1)	1.000000	0.000000
X(DE2, FA4)	32.00000	0.000000
X(DE3, FA2)	11.00000	0.000000
X(DE3, FA7)	40.00000	0.000000
X(DE4, FA6)	5.000000	0.000000
X(DE4, FA8)	38.00000	0.000000
X(DE5, FA1)	34.00000	0.000000
X(DE5, FA2)	7.000000	0.000000
X(DE6, FA3)	22.00000	0.000000
X(DE6, FA6)	27.00000	0.000000
X(DE6, FA7)	3.000000	0.000000

Resultados en LINGO

Row	Slack or Surplus	Dual Price
FO	664.0000	-1.000000
RO(DE1)	0.000000	3.000000
RO(DE2)	22.00000	0.000000
RO(DE3)	0.000000	3.000000
RO(DE4)	0.000000	1.000000
RO(DE5)	0.000000	2.000000
RO(DE6)	0.000000	2.000000
RD(FA1)	0.000000	-4.000000
RD(FA2)	0.000000	-5.000000
RD(FA3)	0.000000	-4.000000
RD(FA4)	0.000000	-3.000000
RD(FA5)	0.000000	-7.000000
RD(FA6)	0.000000	-3.000000
RD(FA7)	0.000000	-6.000000
RD(FA8)	0.000000	-2.000000

Resultados en LINGO




GAMS
FREE TRIAL
SEARCH
Products
Documentation
Download
Consulting Services
Support
Sales
Community
About Us
Best in class mathematical modeling. Performant, scalable, easy to learn.
model - solve - deploy

[GAMS]

```
$title A Transportation Problem (TRNSPORT,SEQ=1)

$onText
This problem finds a least cost shipping schedule that meets
requirements at markets and supplies at factories.

Keywords: linear programming, transportation problem, scheduling
$offText

Set
   i 'canning plants' / seattle,  san-diego /
   j 'markets'        / new-york, chicago, topeka /;

Parameter
   a(i) 'capacity of plant i in cases'
        / seattle    350
          san-diego  600 /

   b(j) 'demand at market j in cases'
        / new-york   325
          chicago    300
          topeka     275 /;

Table d(i,j) 'distance in thousands of miles'
              new-york  chicago  topeka
   seattle         2.5      1.7     1.8
   san-diego       2.5      1.8     1.4;

Scalar f 'freight in dollars per case per thousand miles' / 90 /;

Parameter c(i,j) 'transport cost in thousands of dollars per case';
c(i,j) = f*d(i,j)/1000;
```

```
Variable
   x(i,j) 'shipment quantities in cases'
   z      'total transportation costs in thousands of dollars';

Positive Variable x;

Equation
   cost      'define objective function'
   supply(i) 'observe supply limit at plant i'
   demand(j) 'satisfy demand at market j';

cost..      z =e= sum((i,j), c(i,j)*x(i,j));

supply(i).. sum(j, x(i,j)) =l= a(i);

demand(j).. sum(i, x(i,j)) =g= b(j);

Model transport / all /;

solve transport using lp minimizing z;

display x.l, x.m;
```

Ejemplo

Problema de asignación

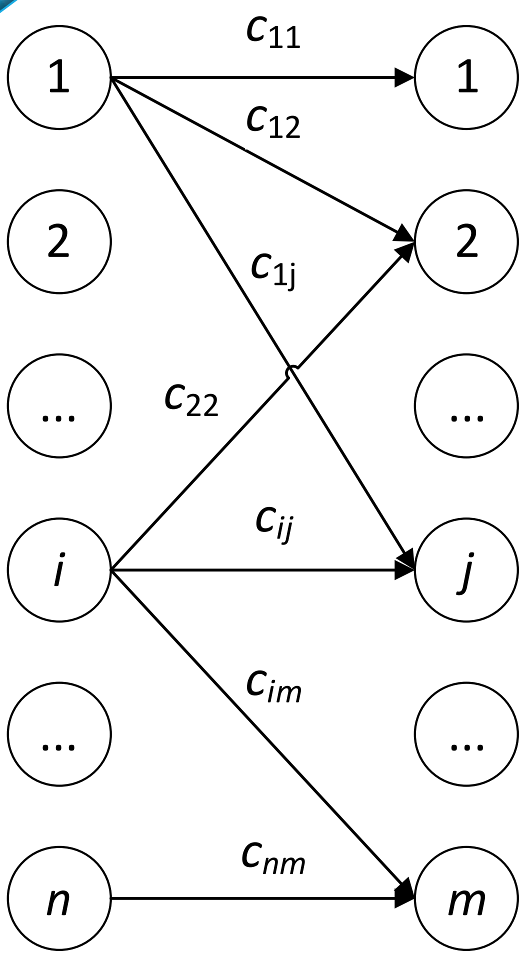
Problema de asignación

En este problema, *m* recursos se asignan a *n* actividades, uno a uno (cada recurso es asignado a lo sumo a una actividad, y cada actividad recibe a lo sumo un recurso). Existe un costo c_{ij} por asignar el recurso *i* a la actividad *j*. Se desea determinar para cada recurso *i* la actividad *j* a la que debe ser asignado con el fin de minimizar el costo total.

Problema de asignación

- Parámetros:
  - m: Cantidad de recursos.
  - n: Cantidad de actividades.
  - c_{ij}: Costo de asignar el recurso i a la actividad j.
- Variables de decisión:
  - x_{ij}: 1 si el recurso i se asigna a la actividad j, si no 0.

Problema de asignación



$$x_{ij} = \begin{cases} 1 & \text{si } i \text{ se asigna a la actividad } j \\ 0 & \text{en caso contrario} \end{cases}$$

Problema de asignación

- Casos:
  - Recursos iguales a las actividades: $m = n$
  - Más recursos que actividades: $m > n$
  - Menos recursos que actividades: $m < n$

Problema de asignación

- Recursos iguales a las actividades: $m = n$
- Función objetivo:
  - Costo ($) = $\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$
- Restricciones:
  - Recurso *i* (persona): $\sum_{j=1}^{n} x_{ij} = 1$
  - Actividad *j* (persona): $\sum_{i=1}^{m} x_{ij} = 1$

Problema de asignación

$$\underset{x_{ij}}{\text{Min}} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

s. a:

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

$$\sum_{i=1}^{m} x_{ij} = 1 \quad \forall j$$

$$x_{ij} \in \{0,1\} \quad \forall i, j$$

$m = n$

Problema de asignación

$$\underset{x_{ij}}{\text{Min}} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

s. a:

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

$$\sum_{i=1}^{m} x_{ij} = 1 \quad \forall j$$

$$x_{ij} \in \{0,1\} \quad \forall i, j$$

$n > m$

Problema de asignación

$$\underset{x_{ij}}{\text{Min}} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\, x_{ij}$$

s. a:

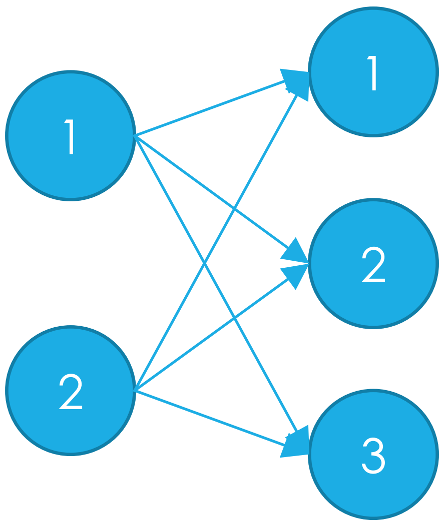
$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i$$

$$\sum_{i=1}^{m} x_{ij} \leq 1 \quad \forall j$$

$$x_{ij} \in \{0,1\} \quad \forall i, j$$

$m < n$

Ejemplo de 2x3



$$\underset{x_{ij}}{\text{Min}} \sum_{i=1}^{2}\sum_{j=1}^{3} c_{ij}x_{ij} = c_{1,1}x_{1,1} + c_{1,2}x_{1,2} + c_{1,3}x_{1,3}$$

$$+c_{2,1}x_{2,1} + c_{2,2}x_{2,2} + c_{2,3}x_{2,3}$$

s. a:

$$x_{1,1} + x_{1,2} + x_{1,3} = 1$$

$$x_{2,1} + x_{2,2} + x_{2,3} = 1$$

$$x_{1,1} + x_{2,1} \leq 1$$

$$x_{1,2} + x_{2,2} \leq 1$$

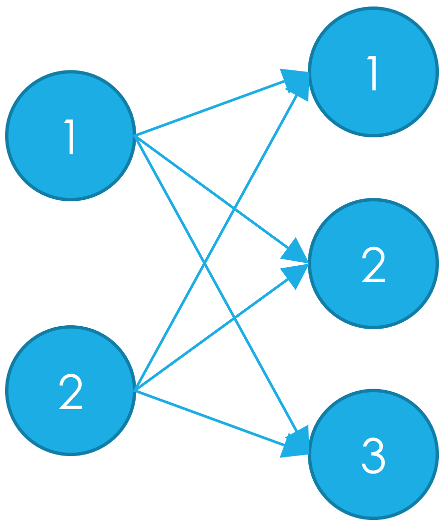
$$x_{1,3} + x_{2,3} \leq 1$$

$$x_{ij} \in \{0,1\} \quad \forall i, j$$

Total unimodularidad

- Una matriz es totalmente unimodular si todas sus submatrices cuadradas tienen determinante 0, 1 o -1. Esta propiedad garantiza que, cuando se tiene un problema de programación lineal (LP) con una matriz de restricciones que es totalmente unimodular y un vector de términos independientes enteros, cualquier vértice (solución básica factible) del poliedro de soluciones será entero.
- No es necesaria la restricción $x_{ij} \in \{0,1\}\ \forall i, \forall j$.

Ejemplo de 2x3



$$\underset{x_{ij}}{\text{Min}} \sum_{i=1}^{2}\sum_{j=1}^{3} c_{ij}\,x_{ij} = c_{1,1}\,x_{1,1} + c_{1,2}\,x_{1,2} + c_{1,3}\,x_{1,3}$$

$$+c_{2,1}\,x_{2,1} + c_{2,2}\,x_{2,2} + c_{2,3}\,x_{2,3}$$

s. a:

$$x_{1,1} + x_{1,2} + x_{1,3} = 1$$

$$x_{2,1} + x_{2,2} + x_{2,3} = 1$$

$$x_{1,1} + x_{2,1} \le 1$$

$$x_{1,2} + x_{2,2} \le 1$$

$$x_{1,3} + x_{2,3} \le 1$$

Líderes de proyectos

Líderes de proyectos

Una empresa está planificando la ejecución de 4 proyectos. Con el fin de lograr la máxima empatía, los equipos deben elegir a sus líderes por votación de un conjunto de 6 candidatos potenciales.

C/E	E1	E2	E3	E4
C1	15	7	4	1
C2	5	8	3	1
C3	5	5	2	10
C4	1	2	2	8
C5	0	1	10	2
C6	0	2	5	2

Líderes de proyectos.lg4

Líderes de proyectos

$$\underset{x_{ij}}{\text{Min}} \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij}\, x_{ij}$$

s. a:

$$\sum_{j=1}^{m} x_{ij} \leq 1 \quad \forall i$$

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

$$x_{ij} \in \{0,1\} \quad \forall i, j$$

$n > m$

```
! Problema de asignación;
! Se desea asignar líderes a equipos de trabajo;
! de forma de maximizar la afinidad;

SETS:
  Candidatos/C1..C6/;
  Equipos/E1..E4/;
  CxE(Candidatos,Equipos):x,a;
ENDSETS

DATA:
! Afinidad a(i,j) del candidato i con el equipo j;
 a = 15   7   4   1
      5   8   3   1
      5   5   2  10
      1   2   2   8
      0   1  10   2
      0   2   5   2;
ENDDATA

! Maximizar la afinidad;
[FO] MAX = @sum(CxE: a*x);

! Asignar un equipo a lo sumo a cada candidato;
@For(Candidatos(i):
  [RC] @sum(Equipos(j): x(i,j)) <= 1
);

! Asignar un candidato a cada equipo;
@For(Equipos(j):
  [RE] @sum(Candidatos(i): x(i,j)) = 1
);
```

Modelo en LINGO

$$\underset{x_{ij}}{\text{Min}} \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij}\, x_{ij}$$

s. a:

$$\sum_{j=1}^{m} x_{ij} \leq 1 \quad \forall i$$

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

$$x_{ij} \in \{0,1\} \quad \forall i, j$$

$n > m$

Lingo Model - Líderes de proyectos

```
! Problema de asignación;
! Se desea asignar líderes a equipos de trabajo;
! de forma de maximizar la afinidad;

SETS:
  Candidatos/C1..C6/;
  Equipos/E1..E4/;
  CxE(Candidatos,Equipos):x,a;
ENDSETS

DATA:
! Afinidad a(i,j) del candidato i con el equipo j;
 a = 15   7   4   1
      5   8   3   1
      5   5   2  10
      1   2   2   8
      0   1  10   2
      0   2   5   2;
ENDDATA

! Maximizar la afinidad;
[FO] MAX = @sum(CxE: a*x);

! Asignar un equipo a lo sumo a cada candidato;
@For(Candidatos(i):
  [RC] @sum(Equipos(j): x(i,j)) <= 1
);

! Asignar un candidato a cada equipo;
@For(Equipos(j):
  [RE] @sum(Candidatos(i): x(i,j)) = 1
);
```

Resultados en LINGO

Variable	Value	Reduced Cost
X(C1, E1)	1.000000	0.000000
X(C2, E2)	1.000000	0.000000
X(C3, E4)	1.000000	0.000000
X(C5, E3)	1.000000	0.000000

Resultados en LINGO

Row	Slack or Surplus	Dual Price
FO	43.00000	1.000000
RC(C1)	0.000000	0.000000
RC(C2)	0.000000	0.000000
RC(C3)	0.000000	0.000000
RC(C4)	1.000000	0.000000
RC(C5)	0.000000	0.000000
RC(C6)	1.000000	0.000000
RE(E1)	0.000000	15.00000
RE(E2)	0.000000	8.000000
RE(E3)	0.000000	10.00000
RE(E4)	0.000000	10.00000