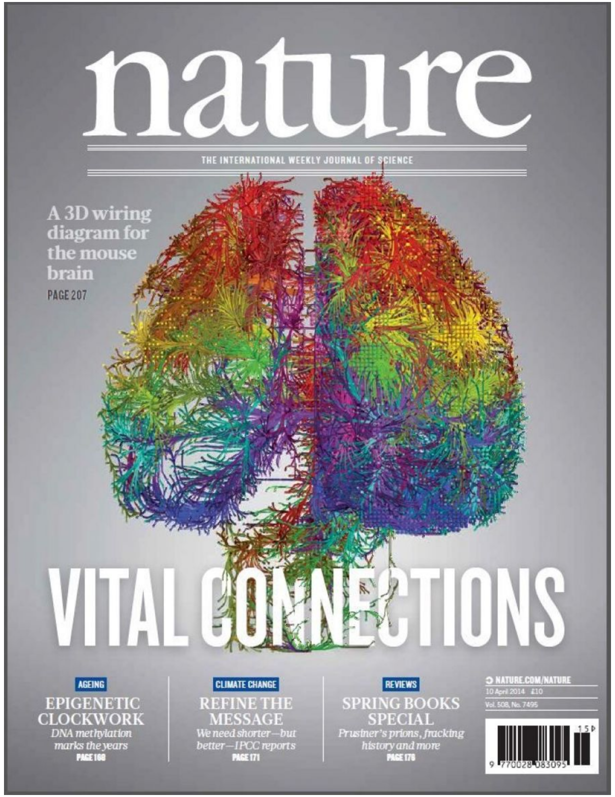
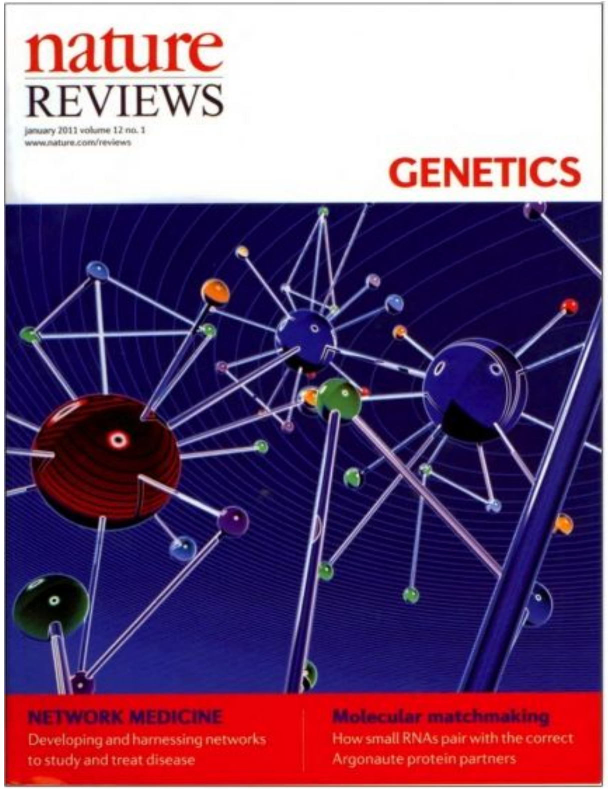
Programación Lineal
Minimum cost flow
Clase 19

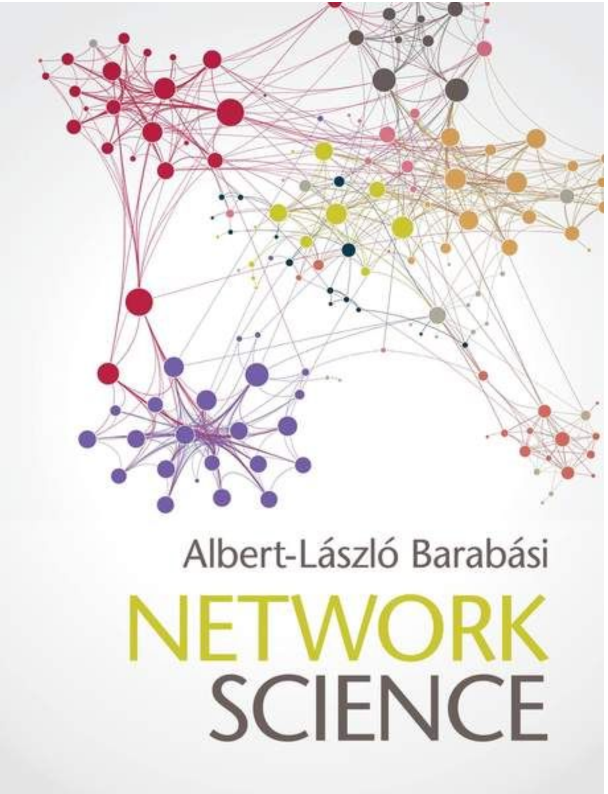
Investigación Operativa UTN FRBA 2021
Curso: I4051
Docente: Martín Palazzo

Teoría de Grafos





Teoría de Grafos: Network Science



El principal libro de teoría de grafos es Network Science, de Albert Barabasi, uno de los investigadores activos mas importantes en el área. El libro es gratuito y de libre acceso desde http://networksciencebook.com/ .

Para consulta de literatura también página 331 a 335 y 368 a 380 del libro Hillier

Elementos de un Grafo

- Nodos (nodes-vertex) -> instancias
- Arcos (edges) -> conectan los nodos

Grafos

- Direccionados (los que vamos a usar en IO)
- Uni-direccionados

Grafos: Matriz de adyacencia

Dada una lista de nodos

$$N = \{n_1, n_2, n_3, \ldots, n_n\}$$

La adyacencia entre nodos se define como

$$A = \begin{cases} A_{ij} = 1 \ \ \exists \text{ conexion entre i j} \\ A_{ij} = 0 \ \ \nexists \text{ conexion entre i j} \end{cases}$$

Adyacencia discreta:

- existe o no conexión

$$A = \begin{cases} A_{ij} = W_{ij} \ \ \exists \text{ conexion e/ i j con peso W} \\ \quad A_{ij} = 0 \qquad\qquad \nexists \text{ conexion e/ i j} \end{cases}$$

Adyacencia continua

- las conexiones están ‘pesadas’

Grafos: Matriz de adyacencia

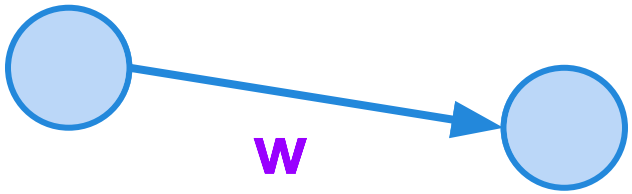
En un grafo direccionado:

- el 1er subíndice de **A** representa el nodo de origen
- el 2do subíndice de **A** representa el nodo destino
- Matriz de adyacencia cuadrada
- Grado de la matriz = cantidad de nodos

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix}$$

Cada posición de la matriz de adyacencia representa un arco (edge) entre nodos

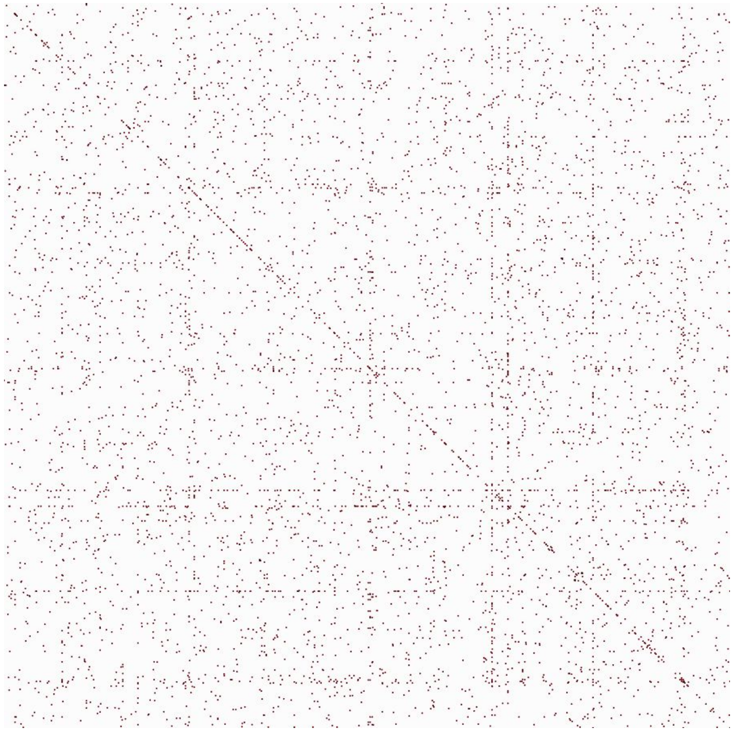
Grafo de un proyecto



El **peso** (weight) de cada arco puede estar caracterizado visualmente con su grosor (y no por su longitud). En la matriz de adyacencia los pesos de los arcos están representados en los elementos de la matriz. Los pesos pueden representar:

- Tiempo
- Distancia
- Flujo
- Resistencia
- Costo
- Interacciones

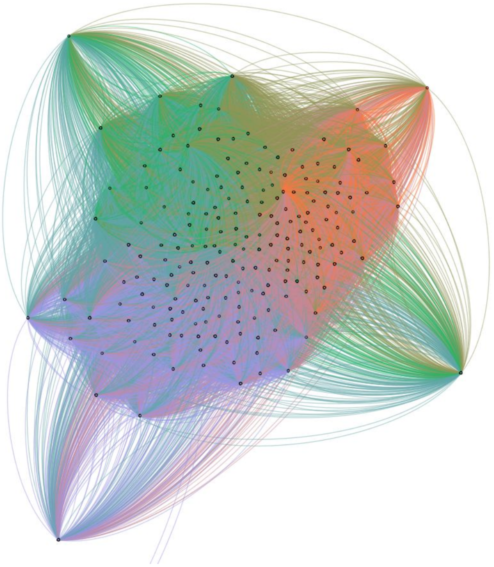
Matriz de adyacencia: biología

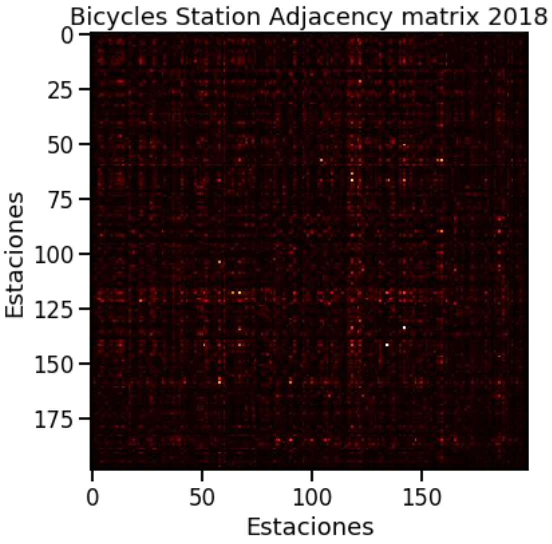


Matriz de adyacencia de un grafo de 2018 nodos. Cada nodo representa una proteína y cada conexión entre nodos representa interacciones entre proteínas.

Fuente http://networksciencebook.com/

Matriz de adyacencia: urbanismo





Matriz de adyacencia de un grafo de 200 nodos y 38483 arcos. Cada nodo representa una estación de BA bici y cada conexión entre nodos representa un viaje realizado entre dichas estaciones.

Este tipo de estudios permite reconocer posibles comunidades de estaciones para poder segmentar la red de bicicletas y poder realizar política pública más eficiente.

Fuente https://medium.com/@martinpalazzo/buenos-aires-bicycle-lanes-iii-d0ca4539e767

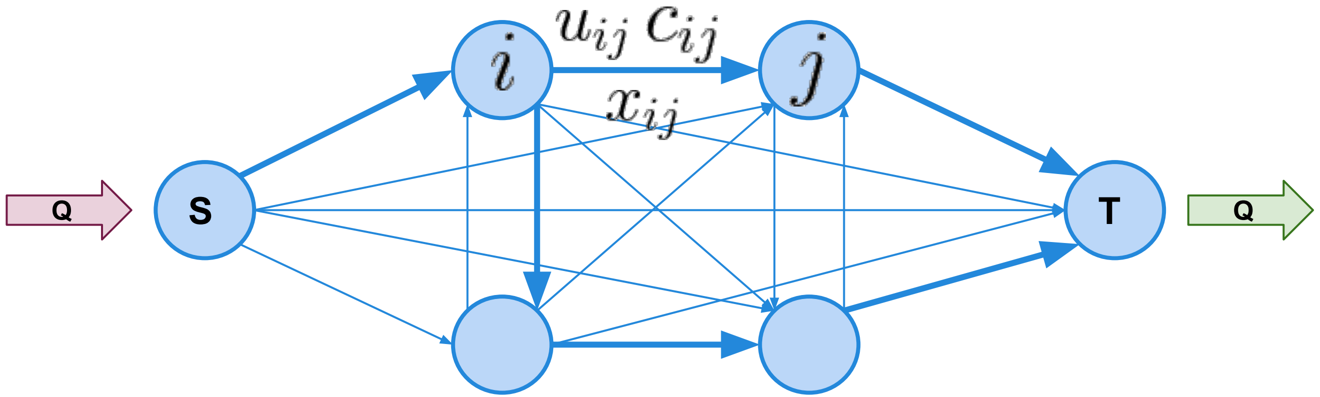
Aplicaciones de Grafos: redes sociales

Un caso de aplicación de grafos es el análisis de redes sociales (digitales, ya que existen las redes sociales fisicas). ¿Qué podrían significar los arcos en este grafo? ¿y que representan sus pesos?

Programación Lineal: Flujo de mínimo costo

Minimum Cost Flow

$$x_{ij} \in \mathbb{R}^+$$



Una red de flujo es una red direccionada con un nodo origen “source” **s** y un nodo destino “target” **t** donde cada arco (i,j) del nodo **i** al nodo **j** tendra

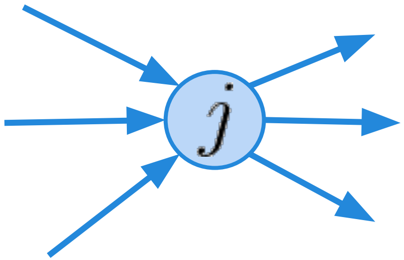
- cantidad de flujo **Xij** que circulara por el arco -> es la variable de decisión (incognita)
- capacidad **Uij** de transportar flujo en el arco (i,j) -> se supone dato
- costo **Cij** de transportar cada unidad de flujo en el arco (i,j) -> se supone dato
- costo total de transportar flujo por un arco sera **Cij * Xij**

En el modelo de red de flujo suponemos una cantidad de flujo **Q** que transportaremos desde el nodo origen **s** al nodo destino **t.** Entonces la **variable de decisión** a calcular para optimizar el problema será la cantidad de flujo Xij que circula por cada arco desde el nodo origen S al nodo destino T de manera tal de **minimizar el costo total de flujo.**

Minimum cost flow: balance de flujo en nodos

$$N = \{n_1, n_2, n_3, ..., n_n\}$$

$$\underbrace{\sum_{k \in N} x_{jk}}_{\text{Flujo saliente}} - \underbrace{\sum_{i \in N} x_{ij}}_{\text{Flujo entrante}} = b_j$$



$$b_j = 0$$

Para cada nodo se podrá determinar el balance entre flujo entrante y flujo saliente representado como Bij. En muchos casos se supondrá que el flujo neto generado/consumido por cada nodo sera Bij = 0, es decir que todo el flujo entrante será igual al saliente (sin contar los nodos origen y destino). El hecho de contemplar Bij = 0 es equivalente a suponer que el modelo tiene una **restricción** de flujo en cada nodo.

Minimum Cost Flow: modelizacion

$$x_{ij} : \text{flujo de nodo i a j}$$

Flujo por arco: Variables de decision

$$z = \sum x_{ij} c_{ij}$$

Costo: Función objetivo

$$0 \leq x_{ij} \leq u_{ij}$$

Capacidad de flujo por arco:
Restriccion 1

$$\sum_{k \in N} x_{jk} - \sum_{i \in N} x_{ij} = b_j \; \forall \; j \in N$$

Balance neto de flujo por nodo:
Restriccion 2

$$x_{ij} \in \mathbb{R}^+$$

Variables de decisión definidas
en los reales positivos

Minimum Cost Flow: modelizacion

$$x_{ij} : \text{flujo de nodo i a j}$$

$$z = \sum x_{ij} c_{ij}$$

$$0 \leq x_{ij} \leq u_{ij}$$

$$\sum_{k \in N} x_{jk} - \sum_{i \in N} x_{ij} = b_j \; \forall \; j \in N$$

$$x_{ij} \in \mathbb{R}^+$$

El problema de optimización se caracteriza por

- Función objetivo lineal
- Restricciones lineales
- Variables de decisión reales positivas

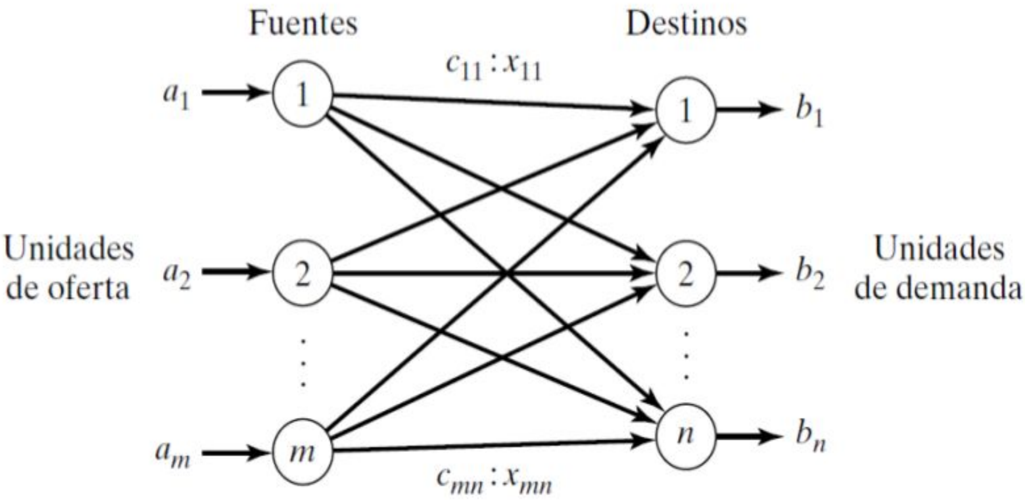
Con estas propiedades definimos al problema de optimización como un problema de programación lineal (estructurado en una red de flujo)

Flujo de mínimo costo: Caso transporte

Programación Lineal

Transporte

Transporte: modelización



Podemos representar el problema con un grafo direccionado donde los nodos de de origen representan las plantas productoras mientras que los nodos destino representan los centros de distribución.

Por otro lado, los arcos representan las rutas posibles entre nodo y nodo. Estarán caracterizados con un "peso" o "costo" por cada unidad de transporte.

Finalmente, la variable a determinar será la matriz X que indica cuánto transportar destino a destino.

Modelización del problema de transporte

Supuestos

- En este modelo lo que deseamos distribuir es **un solo producto**

- **La cantidad total de unidades a distribuir que ofrecen los orígenes es igual a las cantidades totales de unidades que requieren los destinos.**

- Es posible transportar entre todos los orígenes y todos los destinos pero debe incurrir en un **costo unitario** de transportar una unidad entre el origen i y el destino j.

- El objetivo es el transporte de todos los orígenes a todos los destinos **minimizando el costo**

- Hay que decidir cuánto enviar por cada canal (matrix X)

Transporte: caso distribución de cerveza

Supongamos que tenemos el producto Quilmes Cristal que se produce y embotella en la planta Quilmes, Córdoba y Tucuman. En total se producen **300** hecto-litros donde cada planta produce **75**, **125** y **100** respectivamente.

Por otro lado tenemos centros de distribución que abastecen a los clientes de distintas zonas y son Neuquen, Mendoza, Mar del Plata y Santa Fe con demanda **80**, **65**, **70**, **85**.

Cada transporte del destino "i" al "j" tendra un costo asociado Cij. Queremos transportar el 100% del producto minimizando el costo total. Por ende lo que debemos calcular es la cantidad Xij a transportar desde cada destino "i" a cada destino "j".

Transporte: variables de decision

Tres fábricas envían su producto a cuatro distribuidores.

Destino / Origen	Neuquen	Mendoza	MDQ	Sta. Fe	**Oferta**
Quilmes	X11	X12	X13	X14	**75**
Cordoba	X21	X22	X23	X24	**125**
Tucuman	X31	X32	X33	X34	**100**
Demanda	**80**	**65**	**70**	**85**	**300**

Cuanto vale la matriz X???

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{23} & x_{34} \end{bmatrix}$$

Los valores dentro de la tabla representa la cantidad de unidades a transportar.

¿Con qué cantidad se debe confeccionar la orden de compra a cada depósito para que el costo total del pedido sea el mínimo?

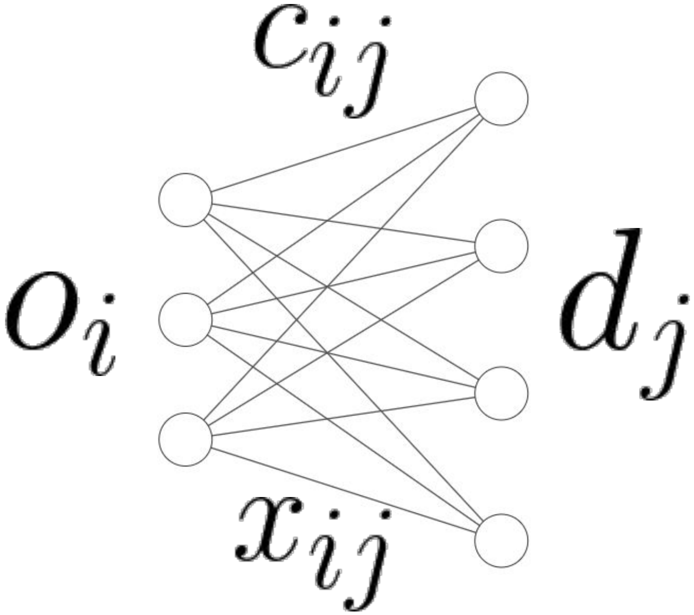
Transporte: matriz de costos

	Costo de embarque ($) por carga				
	Almacén				
	1	2	3	4	Producción
Enlatadora 1	464	513	654	867	75
2	352	416	690	791	125
3	995	682	388	685	100
Asignación	80	65	70	85	

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \end{bmatrix}$$

El problema original podemos estructurarlo en una tabla donde se especifica en el interior de la misma el costo desde la embotelladora "i" al centro de distribución "j". En la columna de la derecha se detalla la oferta que puede hacer cada planta y en el último renglón la demanda de cada centro.

Modelización del problema de transporte



$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{23} & x_{34} \end{bmatrix} \qquad C = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{23} & c_{34} \end{bmatrix}$$

$$O = \begin{bmatrix} o_1 \\ o_2 \\ o_3 \end{bmatrix} \qquad D = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix}$$

- Oi = cantidad a ofertar por nodo i
- Dj = cantidad a demandar por nodo j
- Cij = costo de transporte entre i y j
- **Xij = cantidad a transportar entre i y j**

Modelización del problema de transporte

Matemáticamente se expresa que toda la oferta de los orígenes “a” debe ser igual a toda la demanda de los destinos “b”.

$$\sum_{i=1}^{3} a_i = \sum_{j=1}^{4} b_j$$

Para generalizar podemos decir tanto la oferta como la demanda de todos los orígenes y destinos estará distribuida en la matriz X.

$$\begin{cases} \sum_{i=1}^{4} x_{ij} = a_i \quad , i = (1, 2, 3) \\ \sum_{i=1}^{3} x_{ij} = b_j \quad , j = (1, 2, 3, 4) \end{cases}$$

$$\boxed{Z = \sum_{i=1}^{3} \sum_{i=1}^{4} \mathbf{c}_{ij} \mathbf{x}_{ij}}$$

Por último la función objetivo (funcional) estará caracterizada por la **minimización** del costo total del transporte.

Modelización del problema de transporte

$$\min z = c_{11}x_{11} + c_{12}x_{12} + c_{13}x_{13} + c_{14}x_{14} + c_{21}x_{21} + c_{22}x_{22} + c_{23}x_{23} + c_{24}x_{24} + c_{31}x_{31} + c_{32}x_{32} + c_{33}x_{33} + c_{34}x_{34}$$

$$x_{11} + x_{12} + x_{13} + x_{14} = o_1$$
$$x_{21} + x_{22} + x_{23} + x_{24} = o_2$$
$$x_{31} + x_{32} + x_{33} + x_{34} = o_3$$

$$x_{11} + x_{21} + x_{31} = d_1$$
$$x_{12} + x_{22} + x_{32} = d_2$$
$$x_{13} + x_{23} + x_{33} = d_3$$
$$x_{14} + x_{24} + x_{34} = d_4$$

Modelización del problema de transporte

$$C = \begin{bmatrix} c_{11} = 464 & c_{12} = 513 & c_{13} = 654 & c_{14} = 867 \\ c_{21} = 352 & c_{22} = 416 & c_{23} = 690 & c_{24} = 791 \\ c_{31} = 995 & c_{32} = 682 & c_{23} = 388 & c_{34} = 685 \end{bmatrix}$$

$$O = \begin{bmatrix} o_1 = 75 \\ o_2 = 125 \\ o_3 = 100 \end{bmatrix} \qquad O = \begin{bmatrix} d_1 = 80 \\ d_2 = 65 \\ d_3 = 70 \\ d_4 = 85 \end{bmatrix}$$

Modelización del problema de transporte

$$
\begin{aligned}
\min \text{ z } &= 464x_{11} + 513x_{12} + 654x_{13} + 867x_{14} \\
&+352x_{21} + 416x_{22} + 690x_{23} + 791x_{24} \\
&+995x_{31} + 682x_{32} + 388x_{33} + 685x_{34}
\end{aligned}
$$

$$
\begin{aligned}
x_{11} + x_{12} + x_{13} + x_{14} &= 75 \\
x_{21} + x_{22} + x_{23} + x_{24} &= 125 \\
x_{31} + x_{32} + x_{33} + x_{34} &= 100
\end{aligned}
$$

$$
\begin{aligned}
x_{11} + x_{21} + x_{31} &= 80 \\
x_{12} + x_{22} + x_{32} &= 65 \\
x_{13} + x_{23} + x_{33} &= 70 \\
x_{14} + x_{24} + x_{34} &= 85
\end{aligned}
$$

Solucion transporte en python

```python
# definimos si es un problema de minimizacion o maximizacion
transport_ = LpProblem("Transporte", LpMinimize)

# definimos las variables de decision, el tipo de variable y la cota inferior
x1 = LpVariable('x1', lowBound=0, cat='Continuous')
x2 = LpVariable('x2', lowBound=0, cat='Continuous')
x3 = LpVariable('x3', lowBound=0, cat='Continuous')
x4 = LpVariable('x4', lowBound=0, cat='Continuous')
x5 = LpVariable('x5', lowBound=0, cat='Continuous')
x6 = LpVariable('x6', lowBound=0, cat='Continuous')
x7 = LpVariable('x7', lowBound=0, cat='Continuous')
x8 = LpVariable('x8', lowBound=0, cat='Continuous')
x9 = LpVariable('x9', lowBound=0, cat='Continuous')
x10 = LpVariable('x10', lowBound=0, cat='Continuous')
x11 = LpVariable('x11', lowBound=0, cat='Continuous')
x12 = LpVariable('x12', lowBound=0, cat='Continuous')

# primero agregamos la funcion objetivo
transport_ += 464*x1 + 513*x2 + 654*x3 + 867*x4 + 352*x5 + 416*x6 + 690*x7 + 791*x8 + 995*x9 + 682*x10
+ 388*x11 + 685*x12, "Funcion objetivo"

# luego agregamos restricciones
transport_ += x1 + x2 + x3 + x4 == 75, "oferta 1"
transport_ += x5 + x6 + x7 + x8 == 125, "oferta 2"
transport_ += x9 + x10 + x11 + x12 == 100, "oferta 3"
transport_ += x1 + x5 + x9  == 80, "Demanda 1 "
transport_ += x2 + x6 + x10  == 65, "Demanda 2 "
transport_ += x3 + x7 + x11  == 70, "Demanda 3 "
transport_ += x4 + x8 + x12  == 85, "Demanda 4 "

# Resolver el problema con el solver de PULP
transport_.solve()
```

$$
\begin{aligned}
& 464x_{11} + 513x_{12} + 654x_{13} + 867x_{14} \\
\text{min z} \; = & +352x_{21} + 416x_{22} + 690x_{23} + 791x_{24} \\
& +995x_{31} + 682x_{32} + 388x_{33} + 685x_{34}
\end{aligned}
$$

$$
\begin{aligned}
x_{11} + x_{12} + x_{13} + x_{14} &= 75 \\
x_{21} + x_{22} + x_{23} + x_{24} &= 125 \\
x_{31} + x_{32} + x_{33} + x_{34} &= 100
\end{aligned}
$$

$$
\begin{aligned}
x_{11} + x_{21} + x_{31} &= 80 \\
x_{12} + x_{22} + x_{32} &= 65 \\
x_{13} + x_{23} + x_{33} &= 70 \\
x_{14} + x_{24} + x_{34} &= 85
\end{aligned}
$$

Solucion en python

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{23} & x_{34} \end{bmatrix}$$

$$z_{opt} = 152535$$