M1R1 Teoría de grafos	Clase 2

Text aquí

Text aquí

CENTRO PELAI
933015696
C/Pelai, 8 08001
Barcelona

CENTRO CÒRSEGA
933218377
C/Còrsega, 409
08037 Barcelona

#StucomÉsDemà
www.stucom.com

Grafos

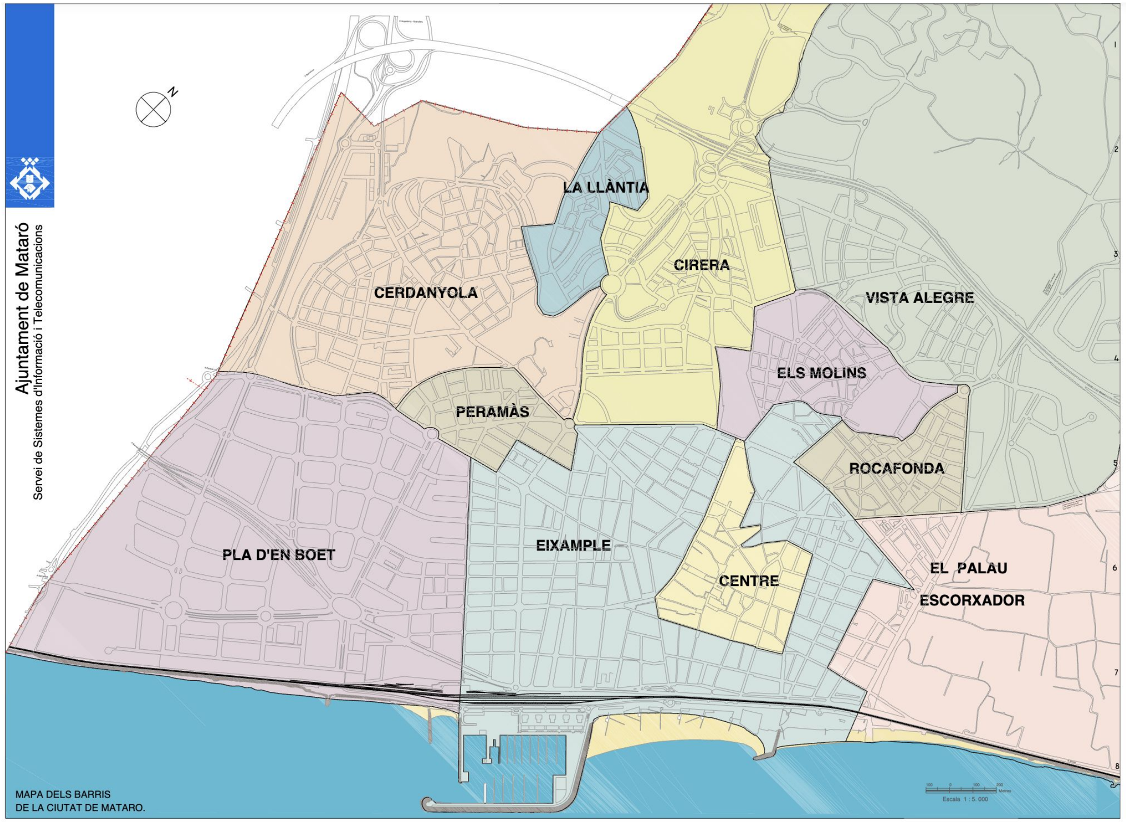
Conjunto de vértices y nodos que se unen por aristas (edges)

G(V,E)

No puede ser que $V \neq \varnothing$
Puede ser que $E \neq \varnothing$

Grafos

Dibujad los barrios de Mataró
Anotad el conjunto de vértices
Anotad el conjunto de aristas



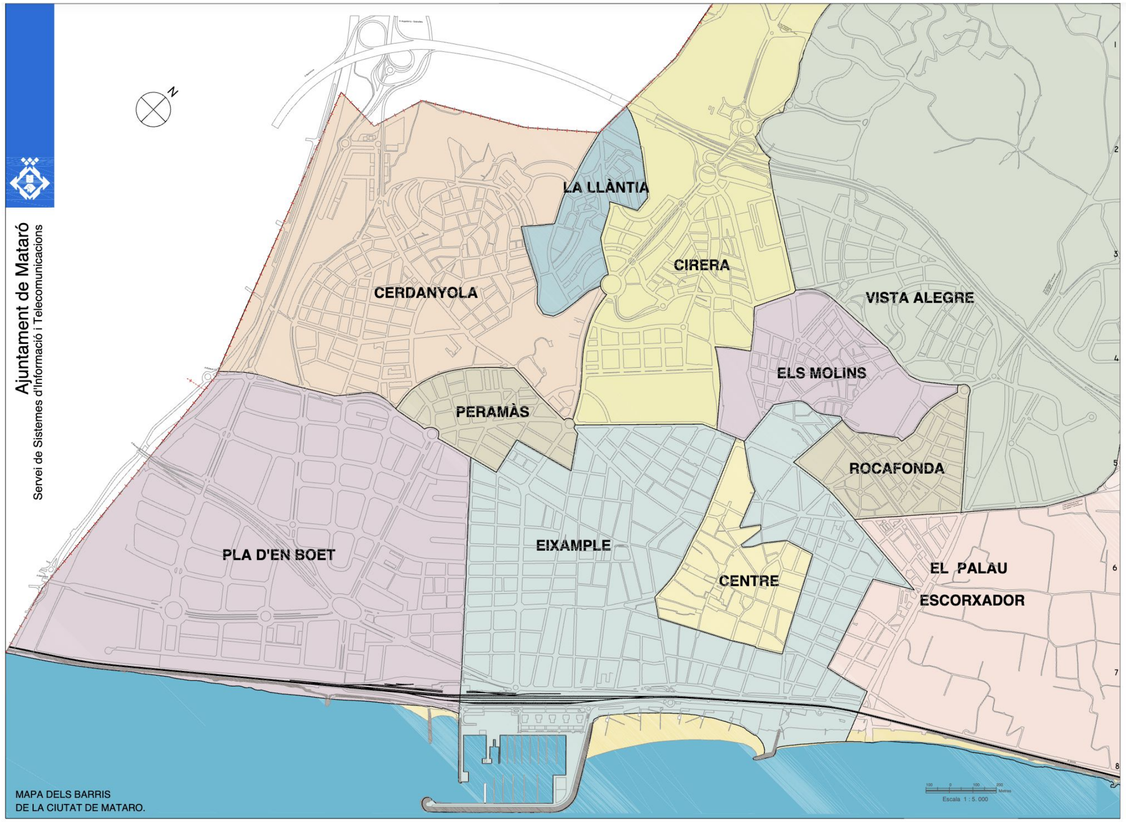
Dibujo Grafo

CENTRO PELAI
933015696
C/Pelai, 8 08001
Barcelona

CENTRO CÒRSEGA
933218377
C/Còrsega, 409
08037 Barcelona

Grafos

Vértice aislado
Aristas con argumentos



Tipos de grafos

Graf **dirigit**

*Eres responsable de distribuir alimentos desde un centro logístico en **Pla d'en Boet** a diferentes barrios de Mataró. Cada barrio está conectado mediante carreteras.*

- ***Nodos:*** *Representan los barrios de Mataró:* ***Pla d'en Boet*** *(centro logístico),* ***Cerdanyola****,* ***Rocafonda****,* ***Centre****,* ***Eixample****,* ***El Palau Escorxador****,* ***Vista Alegre****,* ***Cirera****,* ***La Llàntia****,* ***Els Molins*** *y* ***Peramas****.*
- ***Aristas dirigidas:*** *Representan las rutas disponibles para llegar de un barrio a otro.*
- ***Centro logístico:*** *Nodo* ***Pla d'en Boet****.*

Tipos de grafos

Grafo **dirigido**

Grafo:

- *El centro logístico en **Pla d'en Boet** tiene conexiones directas a **Cerdanyola**, **Centre**, **Peramas**.*
- ***Cerdanyola** está conectada con **Rocafonda**.*
- ***Centre** está conectado con **Eixample** y **El Palau Escorxador**.*
- ***Eixample** está conectado con **Vista Alegre**.*
- ***Rocafonda** tiene conexiones con **La Llàntia**.*
- ***El Palau Escorxador** está conectado con **Els Molins**.*
- ***Vista Alegre** está conectado con **Cirera** y **Els Molins**.*

CENTRO PELAI
933015696
C/Pelai, 8 08001
Barcelona
CENTRO CÒRSEGA
933218377
C/Còrsega, 409
08037 Barcelona

Dibuix Graf

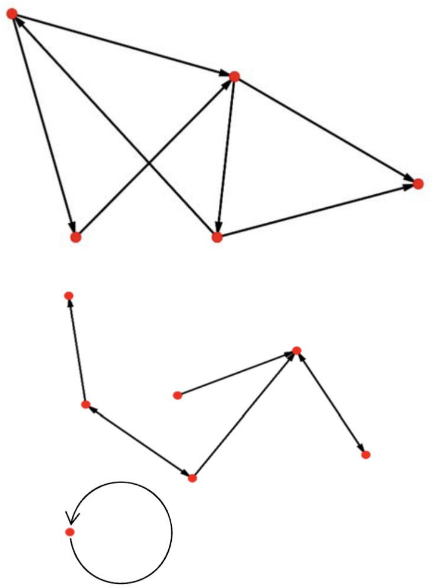
Tipos de grafos

Grafo **dirigido**

- **Arco**
- **Camino:** $\gamma(v_i, v_j \ldots)$
  - *Elemental*
  - *Compuesto*
  - *Ciclo*
- **Bucle:** un vértice conectado consigo mismo
- Grado de **recepción**
- Grado d'**emisión**

Tipos de grafos

- Grado d’**emissión**
- Grado **regular**



Tipos de grafos

Grafo ponderado

Aristas van acompanyades de un **peso** (weight)
$p(v_i, v_j)$

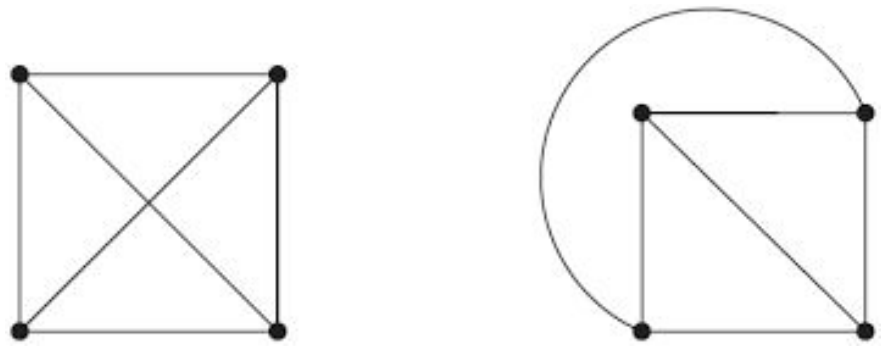
Coste del camíno $\gamma(v_o, v_f)$ es la suma de los pesos totales de cada arista

	Pla d'en Boet	Cerdanyola	Peramàs	La Llàntia	Cirera	Els molins	Vista Alegre	Rocafonda	El palau escorxador	Centre	Eixample
Pla d'en Boet											
Cerdanyola											
Peramàs											
La Llàntia											
Cirera											
Els molins											
Vista Alegre											
Rocafonda											
El palau escorxador											
Centre											
Eixample											

Dibujo Graf

Tipos de grafos

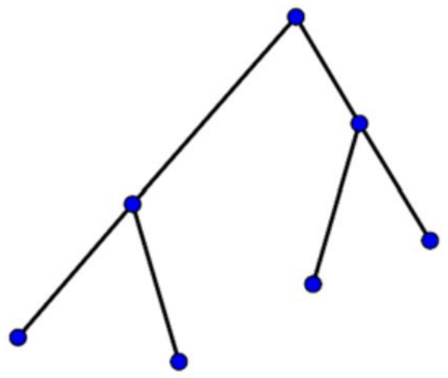
Plano

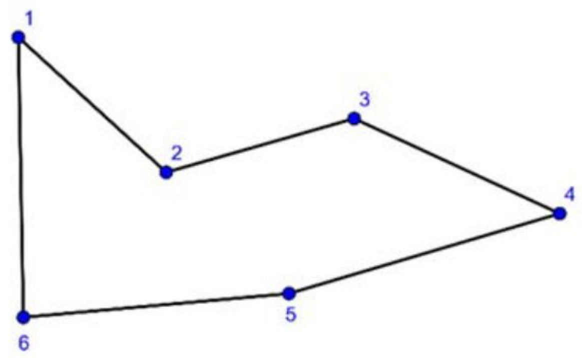


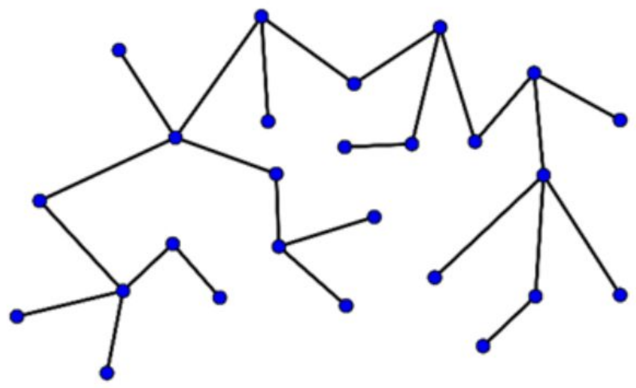
Tipos de grafos

ÁRbol

- Acíclico
- Conectado por un camino







Matrices

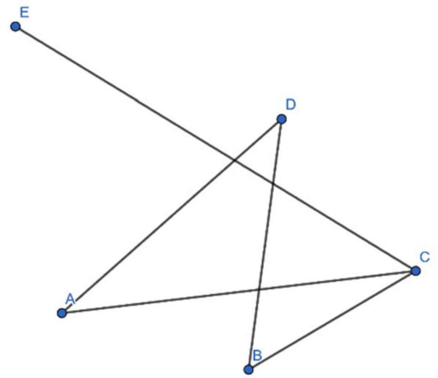
- Matriz **de adjacencia** (0 y 1)

	Pla d'en Boet	Cerdanyola	Peramàs	La Llàntia	Cirera	Els molins	Vista Alegre	Rocafonda	El palau escorxador	Centre	Eixample
Pla d'en Boet											
Cerdanyola											
Peramàs											
La Llàntia											
Cirera											
Els molins											
Vista Alegre											
Rocafonda											
El palau escorxador											
Centre											
Eixample											

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Matrices

- **Matriu de adjecencia**
- **Dimensión?**
- **Tipo de grafo es?**
  - **Dirigido o no dirigido**
  - **Plano o no plano**



Grafos y Phython

https://networkx.org/documentation/latest/tutorial.html

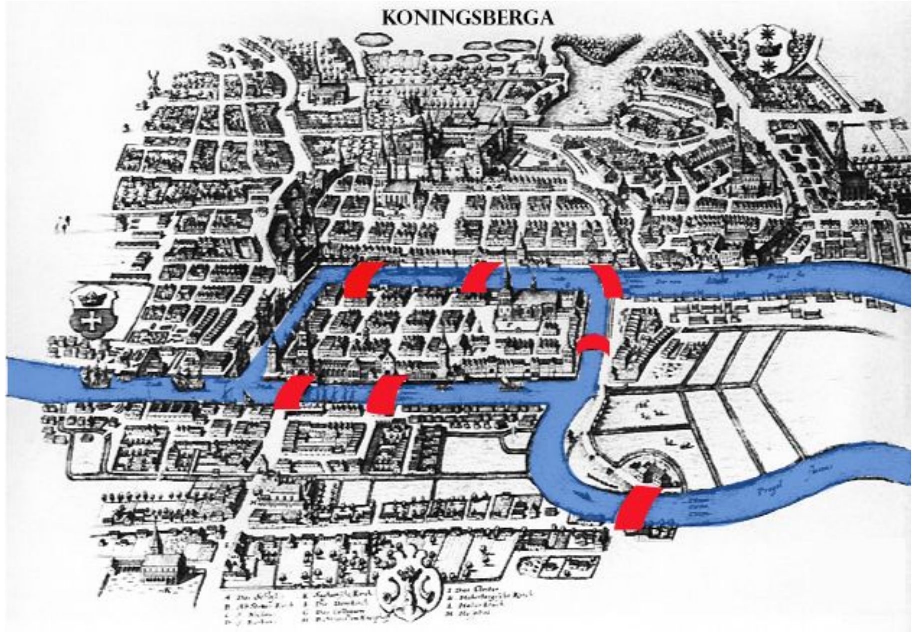
Jupyter notebook in Classroom

Referencia: https://cienciadedatos.net/documentos/pygml01-introduccion-grafos-redes-python

Grafos eulerianos

Los 7 puentes de Königsberg

- *Dado el mapa de Königsberg, como el río Pregel dividiendo el plano en cuatro regiones distinguidas, que están unidas a través de los siete puentes, es posible dar un paseo empezando desde cualquiera de estas regiones, pasando por todos los puentes, recorriendo sólo una vez cada uno, ¿y volviendo al mismo punto de partida?*



Grafos eulerianos

Los 7 puentes de Königsberg

Ciclo euleriano: recórre todas las aristas una sola vez y vuelve al punto de inicio.

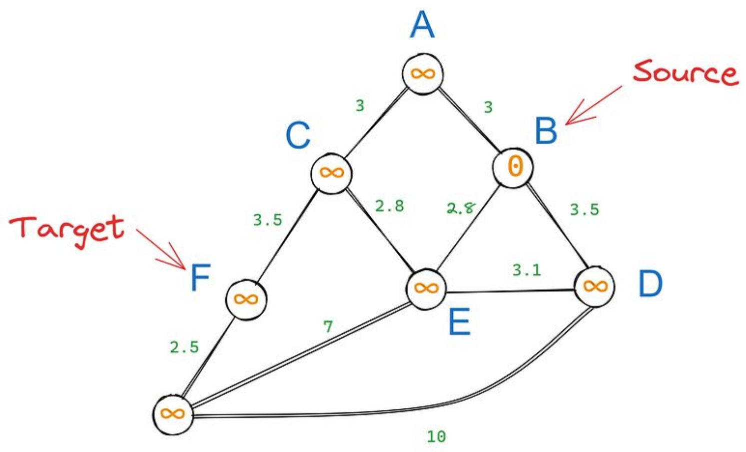
- Màximo 2 vértices imparells
- Todos los vértices tienen grado par

Algoritmo de DIJKSTRA

- Determinar camino más corto
- Explorar todos los caminos más cortos que parten del origen

1. Establecemos nodo origen B.
2. Fijamos las distancias entre B y todos los demás nodos en infinito como los valores de distancia iniciales y provisionales. Fijamos el valor de B a 0, puesto que es la distancia a sí mismo.

Step 1-2: Initialize the graph and node distances



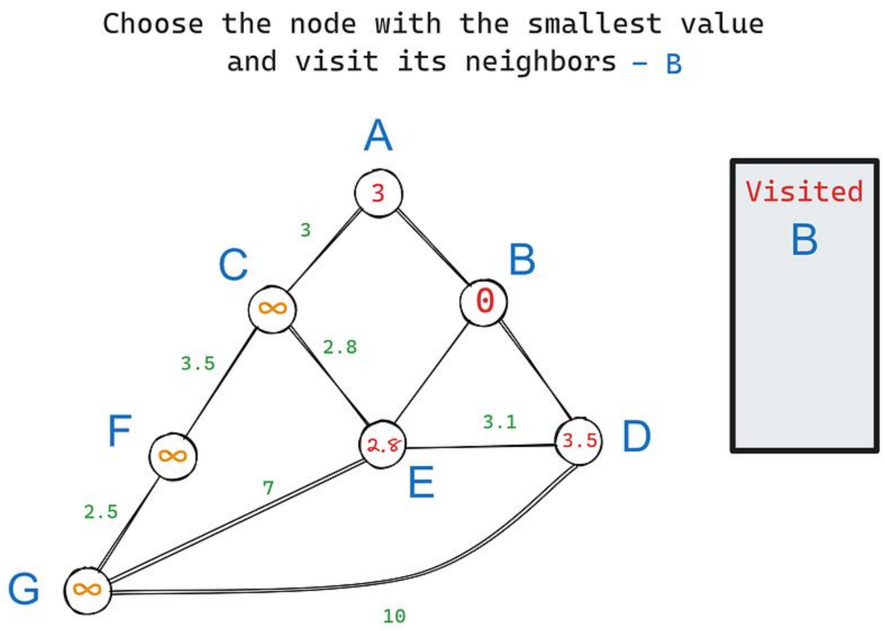
Algoritmo de DIJKSTRA

3. Iterativamente:

a. Elige el nodo con el valor más pequeño como "nodo actual" y visita todos sus nodos vecinos. A medida que visitamos a cada vecino, actualizamos sus valores tentativos desde el infinito hasta los pesos de arista empezando por el nodo origen.

b. Una vez visitados todos los vecinos del nodo actual, marcamos el nodo actual como "vistado". Cuando un nodo está marcado como "vistado", su valor es ya el camino más corto desde el objetivo.

c. El algoritmo vuelve al paso a) y elige el nodo con el menor valor.



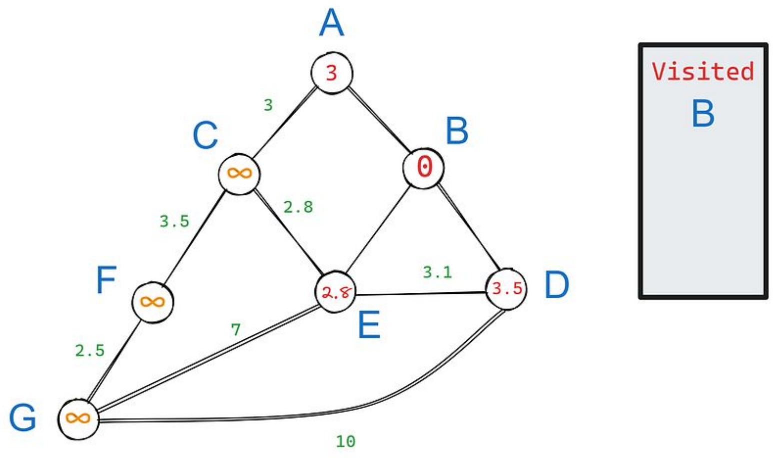
Algoritmo de DIJKSTRA

En nuestro grafo, B tiene tres vecinos: A, D y E. Visitamos cada uno de ellos empezando por el nodo raíz y actualizamos sus valores (iteración 1) en función de los pesos de sus aristas.

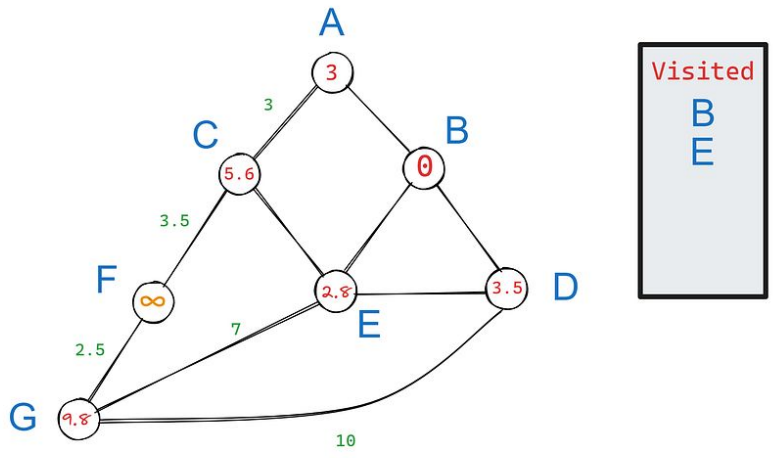
En la iteración 2, volvemos a elegir el nodo con el menor valor, esta vez E. Sus vecinos son C, D y G. B queda excluido porque ya lo hemos visitado. Actualizamos los valores de los vecinos de E.

Fijamos el valor de C en 5,6 porque su valor es la suma acumulada de los pesos de B a C. Lo mismo ocurre con G. Sin embargo, si te fijas, el valor de D sigue siendo 3,5 cuando debería haber sido 3,5 + 2,8 = 6,3, al igual que con los demás nodos.

Choose the node with the smallest value and visit its neighbors – B



Choose the node with the smallest value and visit its neighbors – E



Algoritmo de DIJKSTRA

La regla es que si la nueva suma acumulada es mayor que el valor actual del nodo, no la actualizaremos porque el nodo indica ya la distancia más corta a la raíz. En este caso, el valor actual de D, 3,5 señala ya la distancia más corta a B, porque son vecinos.

Continuamos hasta visitar todos los nodos. Cuando esto ocurra finalmente, tendremos la distancia más corta a cada nodo desde B y podremos buscar simplemente el valor de B a F.

CENTRO PELAI
933015696
C/Pelai, 8 08001
Barcelona

CENTRO CÒRSEGA
933218377
C/Còrsega, 409
08037 Barcelona

Algoritmo de DIJKSTRA

En resumen, los pasos son:

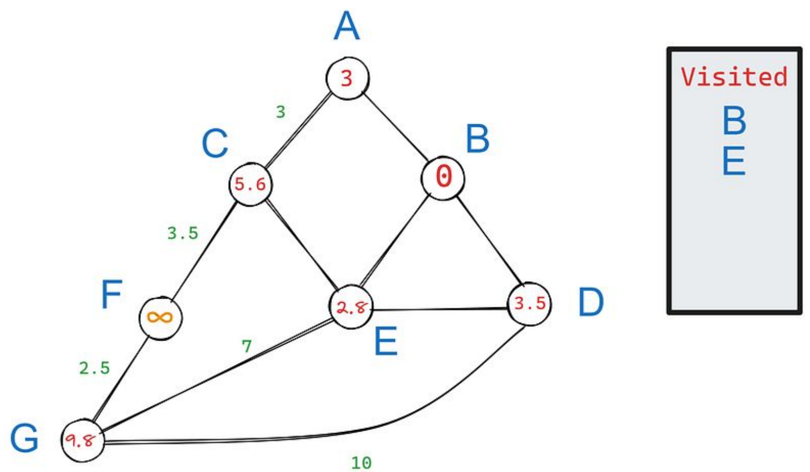
1. Inicializa el grafo con el nodo origen que tomará el valor 0 y todos los demás nodos infinito. Empieza con el origen como "nodo actual".
2. Visita todos los nodos vecinos del nodo actual y actualiza los valores en la suma acumulada de pesos (distancias) desde su origen. Si el valor actual de un vecino es menor que la suma acumulada, permanecerá igual. Marca el "nodo actual" como finalizado.
3. Marca el nodo de valor mínimo inacabado como "nodo actual".
4. Repita los pasos 2 y 3 hasta acabar con todos los nodos.

CENTRO PELAI
933015696
C/Pelai, 8 08001
Barcelona
CENTRO CÒRSEGA
933218377
C/Còrsega, 409
08037 Barcelona

Algoritmo de DIJKSTRA

- ['B', 'E', 'C', 'F']

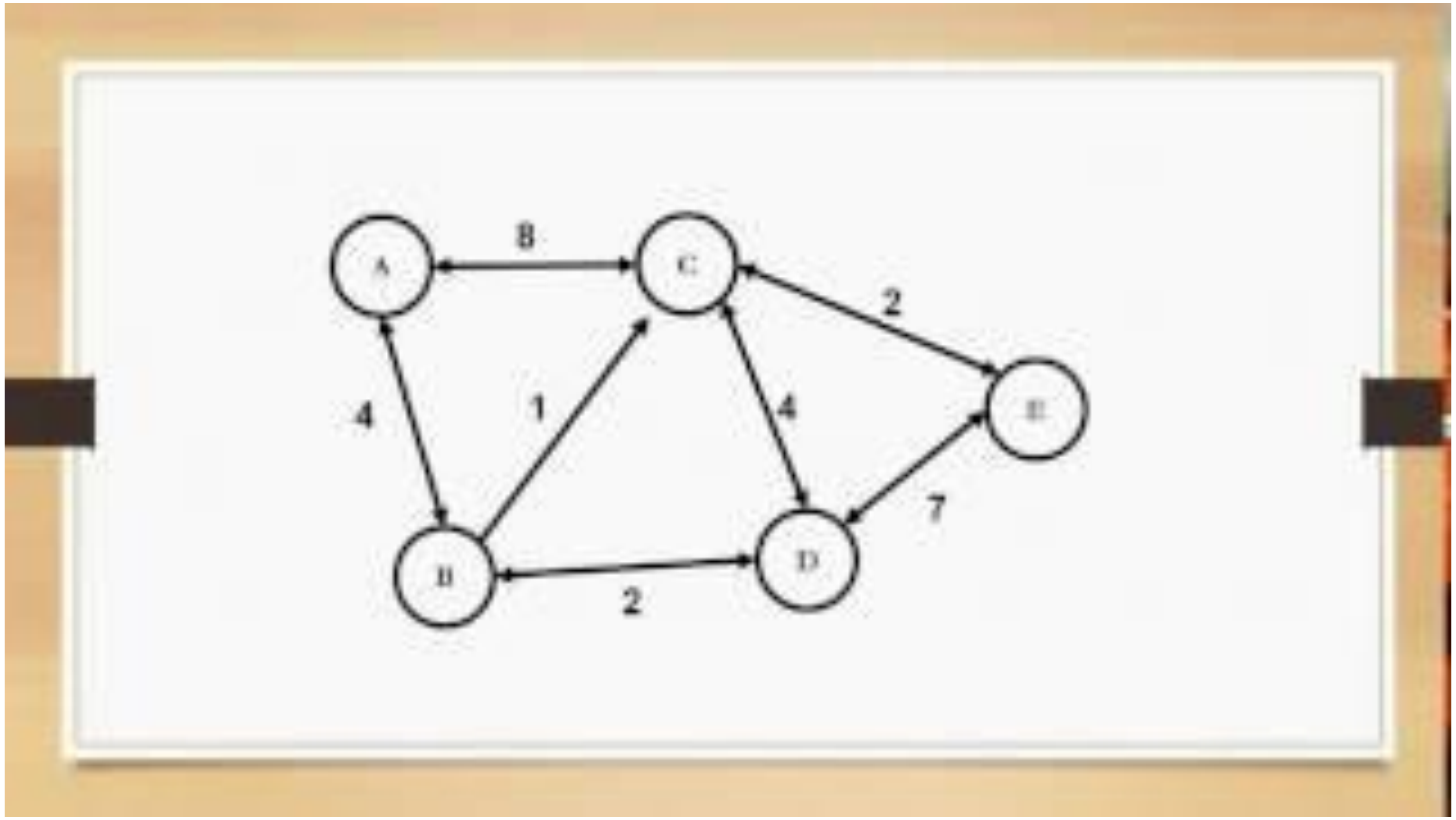
Choose the node with the smallest value
and visit its neighbors – E



Algoritmo de FLOYD-WARSHALL

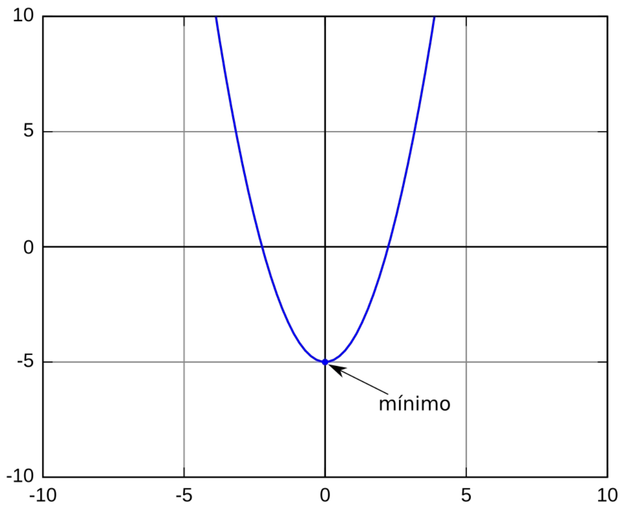
- Determinar camino más corto entre todos los vértices del grafo
- Matriz de distancias entre cada pareja de puntos (si no hay distancia, es infinita)
- Matriz de nodos previos

Algoritmo de FLOYD-WARSHALL



Algoritmos de OPTIMIZACIÓN MULTIOBJETIVA

- En un problema de optimización, se tratará de encontrar una solución que represente el valor óptimo para una función objetivo.
- Un único objetivo: máximo o mínimo
- x = 0 y = - 5



Algoritmos de OPTIMIZACIÓN MULTIOBJETIVA

- El modelado de optimización es un enfoque matemático que se utiliza para encontrar la mejor solución a un problema desde un conjunto de opciones posibles, teniendo en cuenta restricciones y objetivos específicos.
- En la mayoría de casos, los problemas que necesitan optimización tienen más de un objetivo.
- Es difícil encontrar un óptimo de forma simultánea por cada objetivo del problema.
- Existe un conflicto entre objetivos
- La mejora de uno afecta a otro objetivo.
- Una o varias mejores soluciones.

Algoritmos de OPTIMIZACIÓN MULTIOBJETIVA

- Realidad más compleja
- Objetivos: minimizar contaminación, maximizar beneficio, servir a los clientes con el menot tiempo posible, reducir el paro, maximizar ayudas sociales, aumentar sueldos…
- Los objetivos pueden estar en conflicto entre ellos
- Criterios

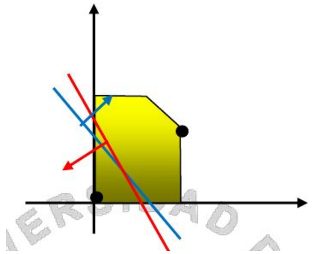
Opt $\{f_1(x), \ldots, f_r(x)\}$ en un conjunto factible
x: variables de decisión
f son las funciones objetivo

Ejemplode OPTIMIZACIÓN MULTIOBJETIVA

- Si los resolvemos gráficamente, podemos comprobar que la soluciones son (20,20) para el problema de maximizar y (0,0) para el de minimizar

-

- Fuente: [Mates II Universidad de Sevilla]



$$\begin{aligned} &Max\ 3x+2y;\quad Min\ 2x+y \\ &s.a.\quad 2x+3y \le 100 \\ &\qquad\quad 0 \le x \le 20 \\ &\qquad\quad 0 \le y \le 25 \end{aligned}$$

Algoritmos de OPTIMIZACIÓN MULTIOBJETIVA

Beneficios, ingresos, eficiencia

Costes, desperdicios, tiempos.

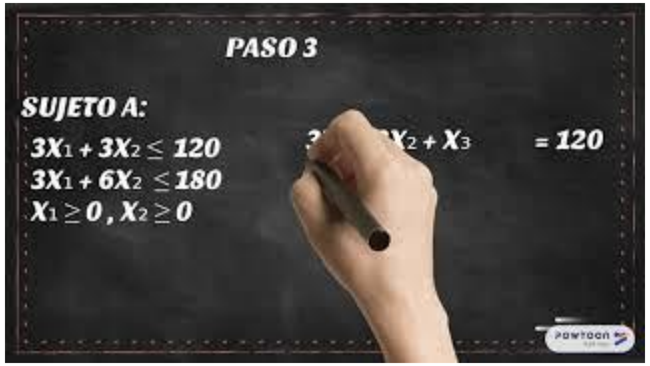
Ejemplode OPTIMIZACIÓN MULTIOBJETIVA

- Una compañía eléctrica puede producir Kw/h mediante el uso de su central eléctrica en cantidad "x" o mediante el uso de sus plantas eólicas en cantidad "y". El beneficio obtenido viene dado por la expresión $f_1(x) = 3x + 2y$. La contaminación producida es $f_2(x) = 2x + y$. Las restricciones técnicas son:

$$2x + 3y \le 100$$
$$0 \le x \le 20$$
$$0 \le y \le 25$$

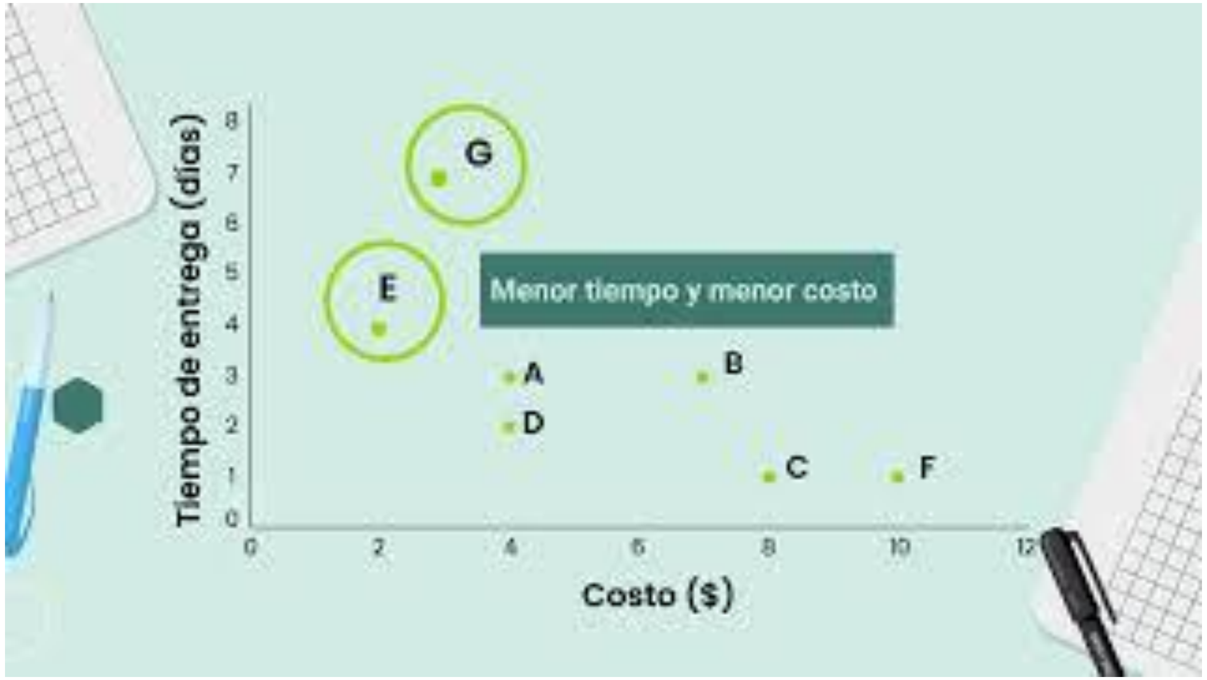
- Los objetivos de la compañía son maximizar el beneficio y minimizar la contaminación producida.
- El problema queda planteado como:
- Fuente: Mates II Universidad de Sevilla

$$\begin{aligned} &Max\ 3x + 2y;\ \ Min\ 2x + y \\ &s.a.\ \ 2x + 3y \le 100 \\ &\qquad 0 \le x \le 20 \\ &\qquad 0 \le y \le 25 \end{aligned}$$


PASO 3
SUJETO A:
3X1 + 3X2 ≤ 120
3X1 + 6X2 ≤ 180
X1 ≥ 0 , X2 ≥ 0
X2 + X3
= 120

Algoritmos de OPTIMIZACIÓN MULTIOBJETIVA

PARETO



Algoritmos de OPTIMIZACIÓN MULTIOBJETIVA

SUMA PONDERADA

Se optimizará el valor obtenido mediante la suma de los valores correspondientes a los distintos objetivos, multiplicados cada uno por un coeficiente de peso. Estos coeficientes de peso establecerán la importancia relativa de cada objetivo.

$$\min \sum_{i=1}^{k} w_i f_i(x)$$

Algoritmos de OPTIMIZACIÓN MULTIOBJETIVA

ASIGNACIÓN DE PRIORIDADES

Estos métodos tienen en común que establecen unas prioridades entre los distintos objetivos, teniendo en cuenta su importancia relativa durante el proceso de optimización.

Método de suma ponderada

PREPARASE LA PRESENTACIÓN

https://es.slideshare.net/slideshow/tema-2-programacin-de-metas-y-optimizacin-multiobjetiva/250991511#1

Grupo 1

Grupo 2

Grupo 3

Grupo 4

Grupo 5

Grupo 6

Grupo 7

PREPARASE LA PRESENTACIÓN

https://es.slideshare.net/slideshow/tema-2-programacin-de-metas-y-optimizacin-multiobjetiva/250991511#1

Grupo 1 - 8 3

Grupo 2 - 10 5

Grupo 3 - 7 4

Grupo 4 - 11 2

Grupo 5 - 6

Grupo 6 - 12

Grupo 7 - 9

Algoritmos de OPTIMIZACIÓN MULTIOBJETIVA

EJEMPLO

Consideremos un escenario hipotético en el que una empresa de entrega, "RapidLogistics", quiere optimizar sus rutas de entrega para una flota de vehículos para minimizar los costos de combustible y al mismo tiempo garantizar entregas oportunas.

1. **Formular el problema** *RapidLogistics quiere minimizar los costes de combustible al entregar paquetes a distintos clientes dentro de una ciudad. Las variables de decisión son las rutas que toma cada vehículo y el objetivo es minimizar el consumo de combustible.*
2. **Identificar restricciones** *Para RapidLogistics, las limitaciones incluyen:*

- *Ventanas temporales: cada cliente tiene una ventana de tiempo específica durante la cual se pueden realizar entregas.*
- *Capacidad del vehículo: cada vehículo tiene una capacidad máxima de peso y volumen para los paquetes.*
- *Debe visitar a todos los clientes: cada cliente debe ser visitado exactamente una vez.*
- *Límite de combustible: la capacidad total de combustible de todos los vehículos combinados es limitada.*

Algoritmos de OPTIMIZACIÓN MULTIOBJETIVA

EJEMPLO

3. Elegir un tipo de modelo de programación *Decida si su problema se puede representar como programación lineal (o optimización lineal), programación no lineal, programación entera, programación cuadrática o algún otro tipo de programación matemática.*

4. Recopilar datos *RapidLogistics debe recopilar datos sobre las ubicaciones de los clientes, sus intervalos de tiempo, los tamaños de los paquetes, las tasas de consumo de combustible de los vehículos, las capacidades de los vehículos y el límite de combustible de todos los vehículos.*

5. Construcción de modelos *En el caso de RapidLogistics, la función objetivo podría ser minimizar la suma del consumo de combustible en todos los vehículos, y las variables de decisión son variables binarias que indican si un vehículo visita a un cliente o no.*

Problema de mochila (knapsack)
pi 200 100 500 700 100
bi 10 40 20 30 50
(1,0,0,1,0)
Capacidad=1000

Algoritmos de optimización multip

Subtítol

Text

Títol

Subtítol

Text

Text	Text	Text	Text
Text			

Gracias