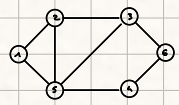
Grafo G é un oggetto matematico composto da 2 insiemi : V = insieme di vertici (o nodi) ed E = insieme di archi

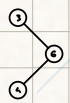
$G = (V, E)$ $V = \{v_1, \dots, v_n\}$

$E = \{ \{v_1, v_2\}, \{v_1, v_3\} \dots \}$



6 nodi , 8 archi

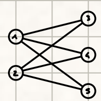
→ grafo non orientato

NB: •  sotto-grafo indotto

•  sotto-grafo non indotto

definizioni :
- 2 vertici sono adiacenti se sono gli estremi di un arco
- 2 archi sono adiacenti se hanno un nodo in comune
- il vicinato di un nodo é l'insieme dei suoi nodi adiacenti
- grado di un vertice :
  - grado = cardinalitá del vicinato
  - grado interno = n° di archi entranti
  - grado esterno = n° di archi uscenti

es: un problema di trasporto é un grafo

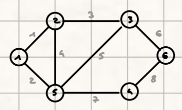


Rappresentazione (con matrici) dei grafi

matrici :
- m. di incidenza $D \in M_{n \times m}$ (n: nodi, m: archi) $d_{ie} = \begin{cases} 1 & e \text{ incidente al nodo } i \\ 0 & \text{altrimenti} \end{cases}$
- m. di adiacenza $A \in M_{n \times n}$ $a_{ij} = \{ 1 \quad i \text{ adiacente a } j$

0 altrimenti

es:



	1	2	3	4	5	6	7	8
D: 1	1	1	0	0	0	0	0	0
2	1	0	1	1	0	0	0	0
3	0	0	1	0	1	1	0	0
4	0	0	0	0	0	0	1	1
5	0	1	0	1	1	0	1	0
6	0	0	0	0	0	1	0	1

→ indica che nodi collega ogni arco

	1	2	3	4	5	6
A: 1	0	1	0	0	1	0
2	1	0	1	0	1	0
3	0	1	0	0	1	1
4	0	0	0	0	1	1
5	1	1	1	1	0	0
6	0	0	1	1	0	0

→ indica con che nodi è collegato ogni nodo (matrice simetrica per la diagonale)

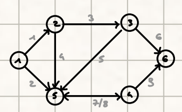
Rappresentazione (con matrici) dei digrafi (= grafi orientati)

matrici:

- m. di incidenza $D \in M_{n \times m}$ (archi) (nodi) $d_{ie} = \begin{cases} 1 & e \text{ è uscente dal nodo } i \\ -1 & e \text{ è entrante } " \\ 0 & \text{altrimenti} \end{cases}$

- m. di adiacenza $A \in M_{n \times n}$ $a_{ij} = \begin{cases} 1 & ; \\ 0 & \text{altrimenti} \end{cases}$

es:



	1	2	3	4	5	6	7	8	9
D: 1	1	1	0	0	0	0	0	0	0
2	-1	0	1	1	0	0	0	0	0
3	0	0	-1	0	1	1	0	0	0
4	0	0	0	0	0	0	1	-1	1
5	0	-1	0	-1	-1	0	-1	1	0
6	0	0	0	0	0	-1	0	0	-1

$$
A: \begin{array}{c|cccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
1 & 0 & 1 & 0 & 0 & 1 & 0 \\
2 & 0 & 0 & 1 & 0 & 1 & 0 \\
3 & 0 & 0 & 0 & 0 & 1 & 1 \\
4 & 0 & 0 & 0 & 0 & 1 & 1 \\
5 & 0 & 0 & 0 & 1 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

→ da dove partono (lato) a dove arrivano (sopra) gli archi

definizione: cammino = sequenza di vertici uniti da un arco con cardinalità = n° di archi (c. "elementare" se non usa 2 volte uno stesso arco; c. "semplice" se non usa lo stesso nodo 2 volte; "ciclo" = cammino con cardinalità ≥ 3 dove origine e destinazione coincidono)

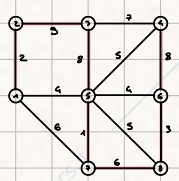
- Hamiltoniano se non usa 2 volte lo stesso nodo
- Euleriano se non usa 2 volte lo stesso arco

Classe di grafi:

- cricca = ogni nodo è collegato con tutti gli altri nodi
- bipartito = ci sono 2 tipi di nodi: origini e destinazioni → prob. di trasporto
- albero = grafico aciclico, si sviluppa solo verso il basso → come l'algoritmo di branch n bound

T è minimamente connesso (= massimamente aciclico, ≡ $m = n - 1$ (n nodi, m archi))

esempio: albero di supporto di costo minimo



→ trovare l'albero di supporto di costo minimo: $c(T) = 26$

→ posso utilizzare una variabile binaria per ogni arco per capire se appartiene all'albero di supporto

$$x_e = \begin{cases} 1 & \text{se appartiene all'albero} \\ 0 & \text{altrimenti} \end{cases} \qquad e \in E \text{ (insieme archi)}$$

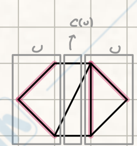
Formulazione 1:

$\min \sum c_e x_e$

st $\sum x_e = n - 1$ $\qquad n = $ n° di nodi

$\sum_{e \in C(U)} x_e \geq 1$ → connesso

↳ vincoli troppo numerosi

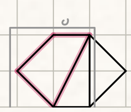


formulazione 2:

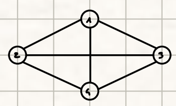
$\min \sum c_e x_e$

st $\sum x_e = n - 1$ $\qquad n = $ n° di nodi

$\sum x_e \leq |U| - 1$ → aciclico



es:



formulazione 1 :

$$\min \; c_{12}x_{12} + c_{13}x_{13} + c_{14}x_{14} + c_{23}x_{23} + c_{24}x_{24} + c_{34}x_{34}$$

$$\text{st} \quad x_{12} + x_{13} + x_{14} + x_{23} + x_{24} + x_{34} = 4 - 1$$

1 $\quad x_{13} + x_{12} + x_{14} \geq 1$

2 $\quad x_{12} + x_{23} + x_{24} \geq 1$

3 $\quad x_{13} + x_{23} + x_{34} \geq 1$

4 $\quad x_{24} + x_{14} + x_{34} \geq 1$

1,2 $\quad x_{13} + x_{23} + x_{14} + x_{24} \geq 1$

1,3 $\quad x_{12} + x_{14} + x_{23} + x_{34} \geq 1$

1,4 $\quad x_{12} + x_{13} + x_{24} + x_{34} \geq 1$

2,3 $\quad x_{12} + x_{13} + x_{24} + x_{34} \geq 1$

2,4 $\quad x_{12} + x_{14} + x_{23} + x_{34} \geq 1$

3,4 $\quad x_{23} + x_{14} + x_{13} + x_{24} \geq 1$

1,2,3 $\quad x_{24} + x_{14} + x_{34} \geq 1$

1,2,4 $\quad x_{13} + x_{23} + x_{34} \geq 1$

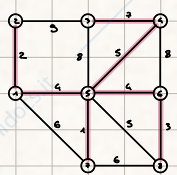
1,3,4 $\quad x_{12} + x_{23} + x_{24} \geq 1$

2,3,4 $\quad x_{13} + x_{12} + x_{14} \geq 1$ $\quad\rightarrow$ troppo dispendioso

algoritmo Kruskal : ordinare per costo crescente gli archi e utilizzare i meno costosi finché posso

tempo $\rightarrow O(m \log m)$ $\quad m$ = n° archi

es:



$C = 26$

algoritmo Prim : provare a estendere sotto-albero connesso con un arco di costo minimo (= scelgo il meno costoso guardando avanti)

tempo $\rightarrow O(n^2)$ $\quad n$ = n° nodi

Problema del cammino minimo (= albero di supporto di costo minimo nei digrafi)

$$\min \; c_{ij}\, x_{ij}$$

archi entranti in i

$$\sum x_{ij} - \sum x_{ki} = \begin{cases} 1 & \text{se } i = s \text{ (partenza)} \\ -1 & \text{se } i = t \text{ (arrivo)} \\ 0 & \text{altrimenti} \end{cases}$$

archi uscenti da i

$\sum_{i,j \in U} x_{ij} \leq card(U) - 1$ → no cicli, condizione inutile se non ho cicli di lunghezza negativa

vedi esercitazione

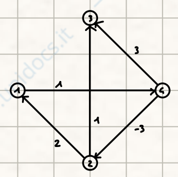
algoritmo di Dijkstra : scelgo il percorso meno costoso tra ogni coppia di punti

tempo → $O(n^2)$ n: n° nodi

NB: funziona solo se i costi sono pos.

algoritmo di Floyd-Warshall (se ho costi negativi)

tempo → $O(n^3)$ n = n° nodi



k = 0

da \ a	1	2	3	4
1	∞	∞	∞	1
2	2	∞	1	∞
3	∞	∞	∞	∞
4	∞	-3	3	∞

k = 1

uso 1 come pivot, fermata intermedia, se costa di più scrivo quello che c'era prima

da \ a	1	2	3	4
1	∞	∞	∞	1
2	2	∞	1	3
3	∞	∞	∞	∞
4	∞	-3	3	∞

k = 2

da \ a	1	2	3	4
1	∞	∞	∞	1
2	2	∞	1	3
3	∞	∞	∞	∞
4	-1	-3	-2	∞

k = 3

da \ a	1	2	3	4
1	∞	∞	∞	1
2	2	∞	1	3
3	∞	∞	∞	∞
4	-1	-3	-2	∞

k = 4

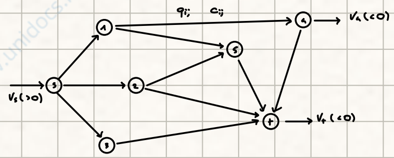
da \ a	1	2	3	4
1	0	-2	-1	1

2	2	0	1	3
3	∞	∞	∞	∞
4	∞	-3	3	0

→ se sulla diag. ho valori neg. ho cicli a lunghezza neg ⇒ distanza non negativa

Problemi di flusso

es: problemi con digrafi in cui ogni arco ha un costo e una capacità massima



q_{ij} = cap. max

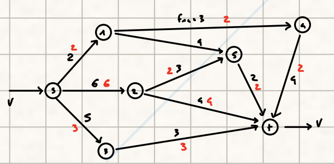
c_{ij} = costo

v_i = flusso (>0 entrate , <0 uscite)

eq^ne di equilibrio : $\sum_{(i,j)} f_{ij} - \sum f_{hi} = v_i$ → quello che esce da i meno quello che entra in i = flusso v

es: problema di flusso massimo

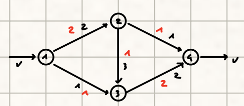
non bado ai costi , calcolo la quantità massima che posso trasportare fino all'arrivo v



$$\sum f_{ij} - \sum f_{hi} = \begin{cases} v & \text{se } i = s \\ -v & \text{se } i = t \\ 0 & \text{per gli altri nodi} \end{cases}$$

$v_{max} = 2 + 6 + 5 - 2 = 11$

es:



$v_{max} = 3$

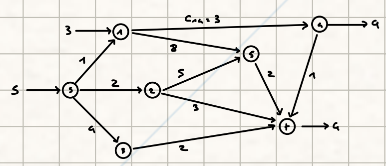
algoritmo Ford - Fulkerson (per flusso max)

tempo → $O(m^2 v_{max})$

creo un digrafo = digrafo originale + archi inversi teorema: max flow, min cut: il valore del flusso massimo è uguale al valore del taglio di minima capacità

↳ si basa sul fatto che il duale del flusso max è il taglio min → se hanno lo stesso valore allora sono entrambi ottimi

es.: problema di flusso di costo minimo



$\min c_{ij} f_{ij} = 1 \cdot 1 + 4 \cdot 3 + 4 \cdot 2 + 4 \cdot 3 = 33$

Problema del commesso viaggiatore = percorso ciclico che passa per ogni vertice con il costo minore

$\min\limits_{H \in \mathcal{H}} d(H)$

NB: per ogni arco ho una variabile $x_{ij} = \begin{cases} 1 & \text{se l'arco appartiene a } H \\ 0 & \text{altrimenti} \end{cases}$

f ob: $\min \sum c_{ij} x_{ij}$

vincoli: $\sum\limits_{j} x_{ij} = 1$ i fisso → archi che escono = 1

$\sum\limits_{i} x_{ij} = 1$ j fisso → archi che entrano = 1

$\sum x_{ij} = n-1$ n = n° nodi