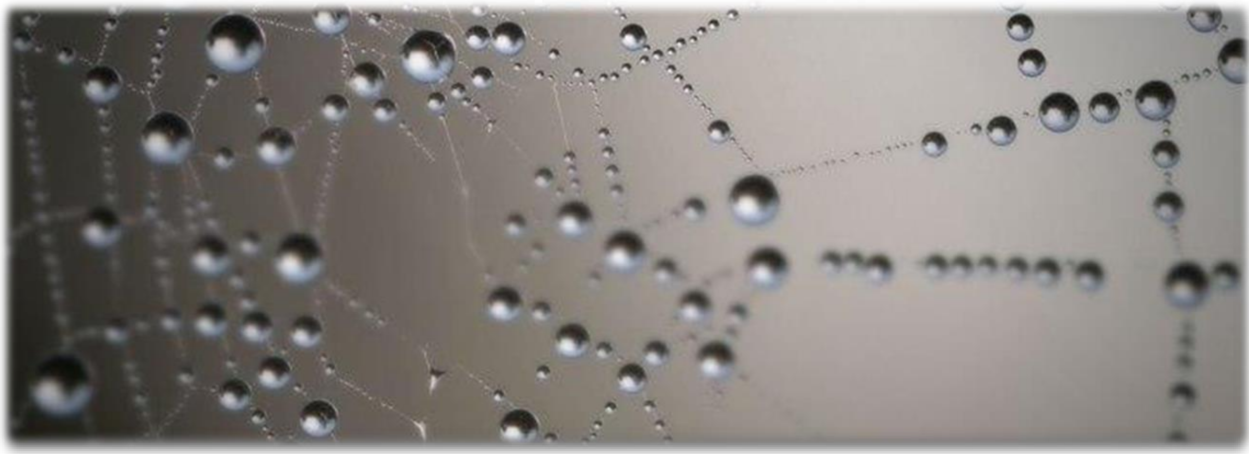
Università Politecnica delle Marche

Introduzione alla Teoria dei Grafi – parte III

ver 2.0.0



Fabrizio Marinelli

fabrizio.marinelli@univpm.it

tel. 071 - 2204823

- Ottimizzazione combinatoria (OC)
- PLI per problemi di OC
- Il minimo albero ricoprente (MST)

[Ipotesi di lavoro] Grafi non orientati e connessi

Problema combinatorio

$U = \{u_1, \ldots, u_n\}$ è un insieme discreto e finito (*insieme universo*)

$$\boxed{(U, \mathfrak{I})}$$

$\mathfrak{I} = \{X \subseteq U : \wp(X)\}$ è la famiglia di sottoinsiemi di U definita dalla proprietà $\wp$ (*regione ammissibile*).

Risolvere un problema combinatorio significa rispondere alla domanda "$\mathfrak{I} \neq \emptyset$?" e, in caso affermativo, esibire un insieme $X \in \mathfrak{I}$.

Problema di ottimizzazione combinatoria

$$\boxed{(U, \Im, f)}$$

$f : U \rightarrow \mathrm{R}$ *funzione peso* associa un numero reale ad ogni elemento di U e ad ogni sottoinsieme X di U

$$\boxed{f(X) = \sum_{u \in X} f(u) \quad \text{per } X \subseteq U}$$

Risolvere $(U, \Im, f)$ significa determinare una soluzione ottima $\underline{Y} \in \Im$, cioè un insieme $\underline{Y}$ tale che

$$f(\underline{Y}) \geq f(X) \qquad \forall\, X \in \Im \qquad \text{(se prob. è di } \textit{massimo}\text{)}$$

Esempi

- Dato un grafo $G = (V, E)$, una funzione peso $w : E \rightarrow \mathbb{R}$ sugli archi e una funzione peso $p : V \rightarrow \mathbb{R}$ sui nodi

Problema combinatorio	U	$\mathfrak{I}$
$\exists$ un Abbinamento ?	E	$\{M \subseteq U\text{: } M$ è un *matching*$\}$
$\exists$ un Insieme Stabile ?	V	$\{S \subseteq U\text{: } S$ è uno *stable-set*$\}$
$\exists$ un Edge-cover ?	E	$\{C \subseteq E\text{: } C$ è un *edge-cover*$\}$
$\exists$ un Node-cover ?	V	$\{T \subseteq U\text{: } T$ è un *node-cover*$\}$

Esempi

- Dato un grafo $G = (V, E)$, una funzione peso $w : E \to R$ sugli archi e una funzione peso $p : V \to R$ sui nodi

Problema di ottimizzazione combinatoria	$f(U)$
Abbinamento (di peso) massimo	$w(M) = \sum_{uv \in M} w(uv)$
Insieme Stabile (di peso) massimo	$p(S) = \sum_{u \in S} p(u)$
Edge-cover (di peso) minimo	$w(C) = \sum_{uv \in C} w(uv)$
Node-cover (di peso) minimo	$p(T) = \sum_{u \in T} p(u)$

- Ottimizzazione combinatoria (OC)
- PLI per problemi di OC
- Il minimo albero ricoprente (MST)

Modelli di PLI: massimo insieme stabile

$S \subseteq V$ è un insieme stabile se $u,v \in S$ implica $\{u,v\} \notin E$.

[Problema] Calcolare $\alpha(G)$ (insieme stabile di massima cardinalità)

Modelli di PLI: massimo insieme stabile …

$S \subseteq V$ è un insieme stabile se $u,v \in S$ implica $\{u,v\} \notin E$.

[Problema] Calcolare $\alpha(G)$ (insieme stabile di massima cardinalità)

$$x_v = \begin{cases} 1 \text{ se } v \in S \\ 0 \text{ altrimenti} \end{cases}$$

STAB

$$\alpha(G) = \max \sum_{u \in V} x_u$$

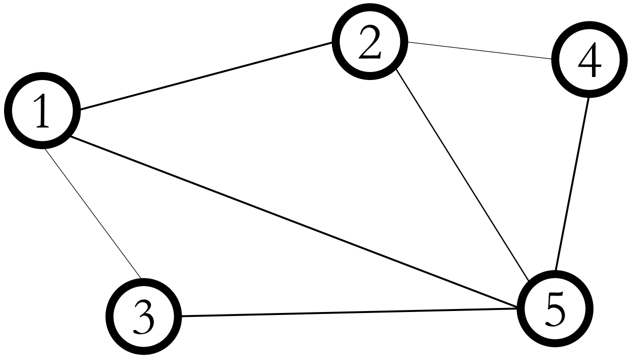
insieme di massima card.

$$x_u + x_v \leq 1 \qquad \forall uv \in E$$

al più un nodo per ogni arco

$$0 \leq x_u \leq 1, \text{intero} \qquad u \in V$$

Modelli di PLI: massimo insieme stabile …



STAB

$$\alpha(G) = \max x_1 + x_2 + x_3 + x_4 + x_5$$
$$x_1 + x_2 \le 1$$
$$x_1 + x_3 \le 1$$
$$x_1 + x_5 \le 1$$
$$x_2 + x_4 \le 1$$
$$x_2 + x_5 \le 1$$
$$x_3 + x_5 \le 1$$
$$x_4 + x_5 \le 1$$
$$0 \le x_1, x_2, x_3, x_4, x_5 \le 1, \text{ intero}$$

- una variabile per nodo
- un vincolo per arco

[Nota] i vincoli $x_v \le 1$ possono essere omessi perché implicati.

massimo insieme stabile: duale …

variabili duali

STAB$_R$

$$
\max x_1 + x_2 + x_3 + x_4 + x_5
$$

$$
\begin{array}{c c}
y_{12}: & \\
y_{13}: & \\
y_{15}: & \\
y_{24}: & \\
y_{25}: & \\
y_{35}: & \\
y_{45}: &
\end{array}
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}
\le
\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}
$$

$$
x_1, x_2, x_3, x_4, x_5 \ge 0
$$

Problema duale

- una variabile per arco
- un vincolo per nodo

$$
\begin{aligned}
\min\ & y_{12}+y_{13}+y_{15}+y_{24}+y_{25}+y_{35}+y_{45} \\
& y_{12} + y_{13} + y_{15} \ge 1 \\
& y_{12} + y_{24} + y_{25} \ge 1 \\
& y_{13} + y_{35} \ge 1 \\
& y_{24} + y_{45} \ge 1 \\
& y_{15} + y_{25} + y_{35} + y_{45} \ge 1 \\
& y_{12}, y_{13}, y_{15}, y_{24}, y_{25} \ge 0
\end{aligned}
$$

Modelli di PLI: copertura con archi …

Reintroducendo le clausole di interezza possiamo dare la seguente interpretazione

$$y_{uv} = \begin{cases} 1 \text{ se } uv \in F \\ 0 \text{ altrimenti} \end{cases}$$

$$\min\ y_{12}+y_{13}+y_{15}+y_{24}+y_{25}+y_{35}+y_{45}$$

insieme di card. minima

$$\left.\begin{array}{r} y_{12} + y_{13} + y_{15} \geq 1 \\ y_{12} + y_{24} + y_{25} \geq 1 \\ y_{13} + y_{35} \geq 1 \\ y_{24} + y_{45} \geq 1 \\ y_{15} + y_{25} + y_{35} + y_{45} \geq 1 \end{array}\right\}$$

almeno un arco per ogni nodo

$$0 \leq y_{12}, y_{13}, y_{15}, y_{24}, y_{25} \leq \boxed{1}$$

i vincoli $y_{uv} \leq 1$ possono essere introdotti perché implicati.

Modelli di PLI: copertura con archi

$F \subseteq E$ è una copertura con archi se $\forall u \in V \ \exists\ uv \in F$.

[Problema] Calcolare $\rho(G)$ (edge-cover di cardinalità minima)

Modelli di PLI: copertura con archi …

$F \subseteq E$ è una copertura con archi se $\forall u \in V \ \exists\, uv \in F$.

[Problema] Calcolare $\rho(G)$ (edge-cover di cardinalità minima)

$$y_{uv} = \begin{cases} 1 \text{ se } uv \in F \\ 0 \text{ altrimenti} \end{cases}$$

EDGE-C

$$\rho(G) = \min \sum_{uv \in E} y_{uv}$$

insieme di card. minima

$$\sum_{v \in N(u)} y_{uv} \geq 1 \qquad \forall u \in V$$

almeno un arco per ogni nodo

$$0 \leq y_{uv} \leq 1, \text{intero} \quad uv \in E$$

Insieme stabile e copertura con archi

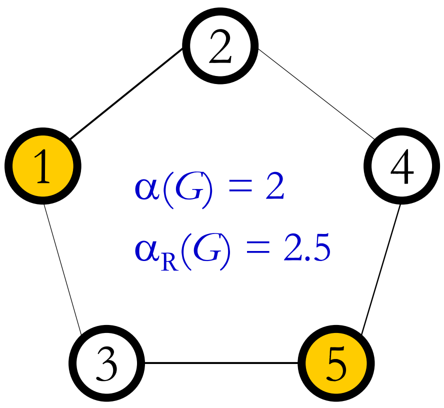
- **[Osservazione]** STAB_R e EDGE-C_R (cioè i *rilassamenti continui* dei modelli STAB e EDGE-C) costituiscono una coppia primale-duale di problemi di PL.

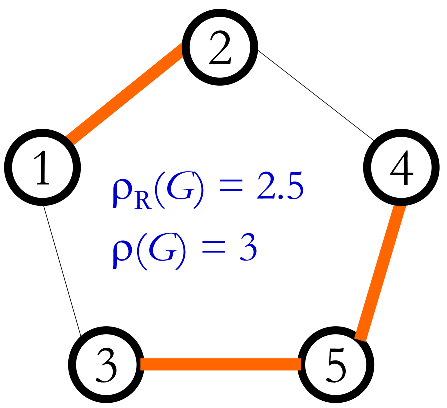
Siano $\alpha_R(G)$ e $\rho_R(G)$ i valori ottimi di STAB_R e EDGE-C_R

$$\alpha_R(G) = \rho_R(G) \qquad \text{(dualità forte)}$$

$$\alpha(G) \le \alpha_R(G) = \rho_R(G) \le \rho(G) \qquad \text{(prop. dei rilassamenti)}$$

Esiste almeno un caso con $\alpha(G) < \rho(G)$





Giochiamo con qualche grafo

- Genera un insieme di grafi random di tipo Erdos-Renyi, ognuno con 150 nodi, e con probabilità di generazione dell'arco $p \in \{0.10, 0.15, 0.20, \ldots, 0.95\}$;

- Usa il solver CPLEX per calcolare un massimo insieme stabile su ognuno di questi grafi;

- Per ogni grafo, stampa $\alpha(G)$ e il running time

- Per ogni grafo, stampa la *best solution* e il gap di dualità dopo 30 secondi di running time

Modelli di PLI: massimo abbinamento

$M \subseteq E$ è un abbinamento se uv, $hw \in M$ implica $u \neq v \neq h \neq w$.

[Problema] Calcolare $\mu(G)$ (abbinamento di massima cardinalità)

Modelli di PLI: massimo abbinamento …

$M \subseteq E$ è un abbinamento se uv, $hw \in M$ implica $u \neq v \neq h \neq w$.

[Problema] Calcolare $\mu(G)$ (abbinamento di massima cardinalità)

$$x_{uv} = \begin{cases} 1 \text{ se } uv \in M \\ 0 \text{ altrimenti} \end{cases}$$

MATCH

$$\mu(G) = \sum_{uv \in E} x_{uv}$$

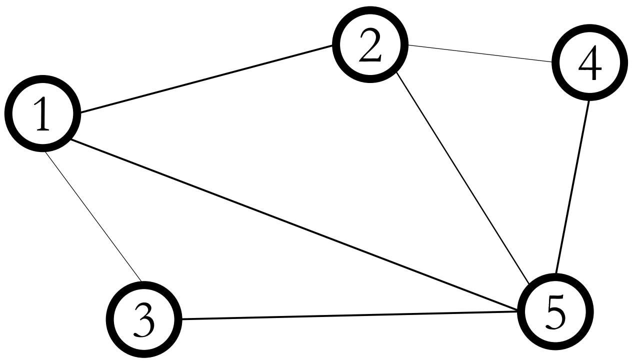
insieme di massima card.

$$\sum_{uv \in \delta(u)} x_{uv} \leq 1 \qquad \forall u \in V$$

al più un arco per ogni nodo

$$0 \leq x_{uv} \leq 1, intero \qquad \forall uv \in E$$

Modelli di PLI: massimo abbinamento …



MATCH

$$\mu(G) = \max x_{12}+x_{13}+x_{15}+x_{24}+x_{25}+x_{35}+x_{45}$$

$$x_{12} + x_{13} + x_{15} \leq 1$$

$$x_{12} + x_{24} + x_{25} \leq 1$$

$$x_{13} + x_{35} \leq 1$$

$$x_{24} + x_{45} \leq 1$$

$$x_{15} + x_{25} + x_{35} + x_{45} \leq 1$$

$$0 \leq x_{12}, x_{13}, x_{15}, x_{24}, x_{25}, x_{35}, x_{45} \leq 1, \text{ intero}$$

- una variabile per arco
- un vincolo per nodo

[Nota] i vincoli $x_{uv} \leq 1$ possono essere omessi perché implicati.

massimo abbinamento: duale …

variabili duali

Problema duale

- una variabile per nodo
- un vincolo per arco

MATCH$_R$

$$\max x_{12}+x_{13}+x_{15}+x_{24}+x_{25}+x_{35}+x_{45}$$

$$\begin{array}{c} y_1: \\ y_2: \\ y_3: \\ y_4: \\ y_5: \end{array} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_{12} \\ x_{13} \\ x_{15} \\ x_{24} \\ x_{25} \\ x_{35} \\ x_{45} \end{bmatrix} \le \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$x_{12}, x_{13}, x_{15}, x_{24}, x_{25}, x_{35}, x_{45} \ge 0$$

$$\min y_1 + y_2 + y_3 + y_4 + y_5$$
$$y_1 + y_2 \ge 1$$
$$y_1 + y_3 \ge 1$$
$$y_1 + y_5 \ge 1$$
$$y_2 + y_4 \ge 1$$
$$y_2 + y_5 \ge 1$$
$$y_3 + y_5 \ge 1$$
$$y_4 + y_5 \ge 1$$
$$y_1, y_2, y_3, y_4, y_5 \ge 0$$

Modelli di PLI: copertura con nodi …

Reintroducendo le clausole di interezza possiamo dare la seguente interpretazione

$$y_u = \begin{cases} 1 \text{ se } u \in T \\ 0 \text{ altrimenti} \end{cases}$$

$$\min y_1 + y_2 + y_3 + y_4 + y_5$$

insieme di card. minima

$$\left.\begin{aligned} y_1 + y_2 &\geq 1 \\ y_1 + y_3 &\geq 1 \\ y_1 + y_5 &\geq 1 \\ y_2 + y_4 &\geq 1 \\ y_2 + y_5 &\geq 1 \\ y_3 + y_5 &\geq 1 \\ y_4 + y_5 &\geq 1 \end{aligned}\right\}$$

almeno un nodo per ogni arco

$$0 \leq y_1, y_2, y_3, y_4, y_5 \leq \boxed{1}$$

i vincoli $y_u \leq 1$ possono essere introdotti perché implicati.

Modelli di PLI: copertura con nodi

$T \subseteq V$ è un trasversale se $\forall uv \in E \;\; u \in T$ oppure $v \in T$.

[Problema] Calcolare $\tau(G)$ (vertex-cover di cardinalità minima)

Modelli di PLI: copertura con nodi …

$T \subseteq V$ è un trasversale se $\forall uv \in E \;\; u \in T$ oppure $v \in T$.

[Problema] Calcolare $\tau(G)$ (vertex-cover di cardinalità minima)

$$y_u = \begin{cases} 1 \text{ se } u \in T \\ 0 \text{ altrimenti} \end{cases}$$

VERTEX-C

$$\tau(G) = \min \sum_{u \in V} y_u$$

insieme di card. minima

$$y_u + y_v \geq 1 \qquad \forall uv \in E$$

almeno un nodo per ogni arco

$$0 \leq y_u \leq 1, \text{intero} \quad u \in V$$

Abbinamento e copertura con nodi

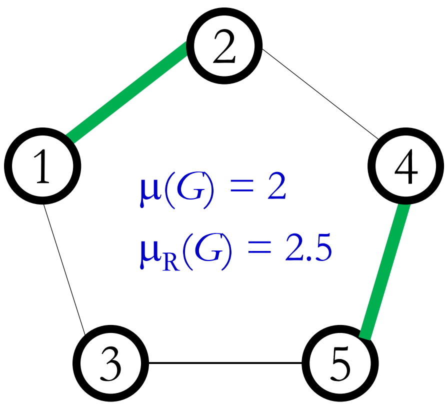
- **[Osservazione]** MATCH$_R$ e VERTEX-C$_R$ (cioè i *rilassamenti continui* dei modelli MATCH e VERTEX-C) costituiscono una coppia primale-duale di problemi di PL.

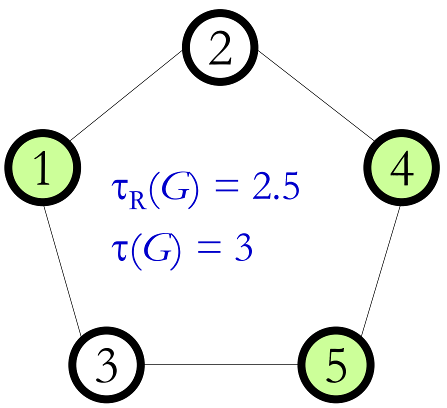
Siano $\mu_R(G)$ e $\tau_R(G)$ i valori ottimi di MATCH$_R$ e VERTEX-C$_R$

$$\mu_R(G) = \tau_R(G) \qquad \text{(dualità forte)}$$

$$\mu(G) \le \mu_R(G) = \tau_R(G) \le \tau(G) \qquad \text{(prop. dei rilassamenti)}$$

Esiste almeno un caso con $\mu(G) < \tau(G)$





Relazioni su grafi bipartiti

[Teorema] (*König*, 1931)

Se G è bipartito allora $\mu(G) = \tau(G)$

[Corollario]

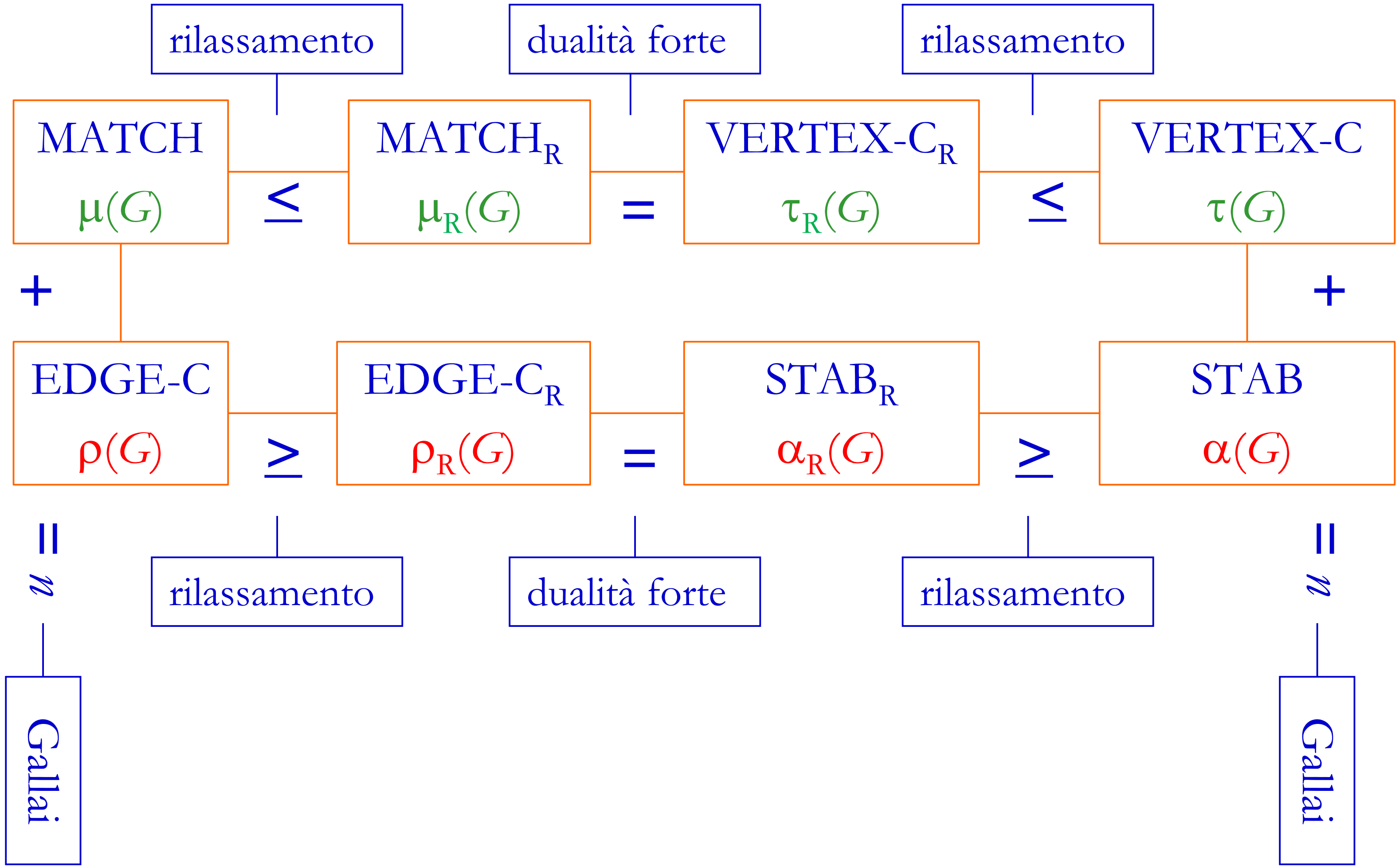
Se G è bipartito allora $\alpha(G) = \rho(G)$

teorema di Gallai

$$\mu(G) + \rho(G) = n$$
$$\tau(G) + \alpha(G) = n$$

$$\Rightarrow \quad \cancel{\mu(G)} + \rho(G) = \cancel{\tau(G)} + \alpha(G)$$

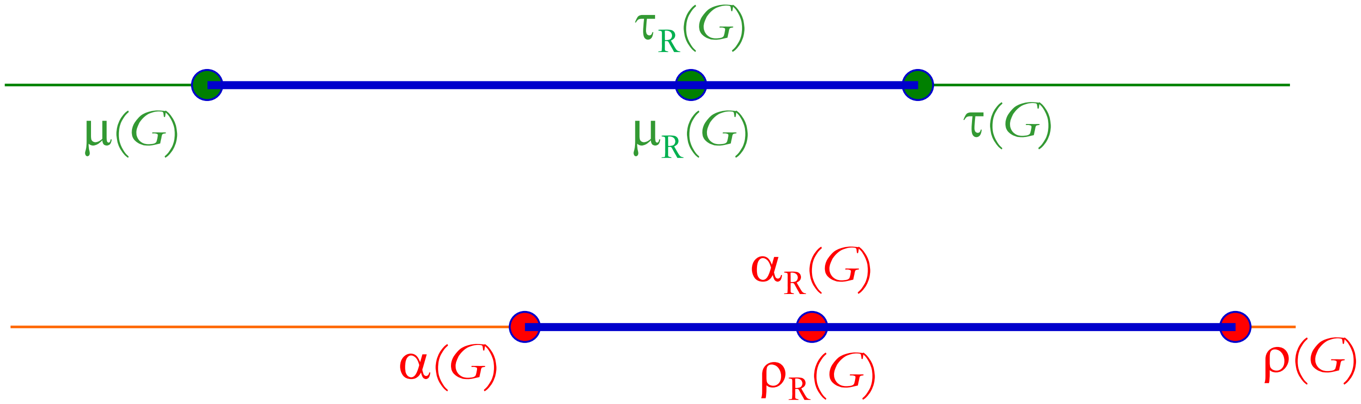
Schema riassuntivo



Schema riassuntivo

$$\mu(G) + \rho(G) = \tau(G) + \alpha(G) = n$$

$$\rho(G) - \alpha(G) = \tau(G) - \mu(G)$$



Esercizi

- Esprimere i problemi precedenti in termini di *set-packing*, *set-covering* o *set-partitioning*

- Stabilire se la seguente formulazione intera modella correttamente il problema di abbinamento massimo. In caso contrario individuare l'errore.

$$A)\quad \max \sum_{uv \in E} x_{uv}$$

$$x_{uv} + x_{vw} + x_{wu} \leq 1 \qquad \forall\, uv, vw, wu \in E$$

$$0 \leq x_{uv} \leq 1, \text{intero} \qquad uv \in E$$

Esercizi

- Stabilire se la seguente formulazione intera modella correttamente il problema di abbinamento massimo. In caso contrario individuare l'errore.

$$B) \quad \max \sum_{uv \in E} x_{uv}$$

$$x_{uv} + x_{vw} \leq 1 \qquad \forall v \in V, uv, vw \in E$$

$$0 \leq x_{uv} \leq 1, \text{intero} \qquad uv \in E$$

Esercizi

- Stabilire se la seguente formulazione intera modella correttamente il problema di abbinamento massimo. In caso contrario individuare l'errore.

$$
\begin{aligned}
C)\quad & \max \sum_{uv \in E} x_{uv} \\
& \sum_{w \in N(u)} x_{uw} + \sum_{w \in N(v)} x_{vw} \leq 1 \qquad \forall uv \in E \\
& 0 \leq x_{uv} \leq 1, \text{intero} \qquad uv \in E
\end{aligned}
$$

Esercizi

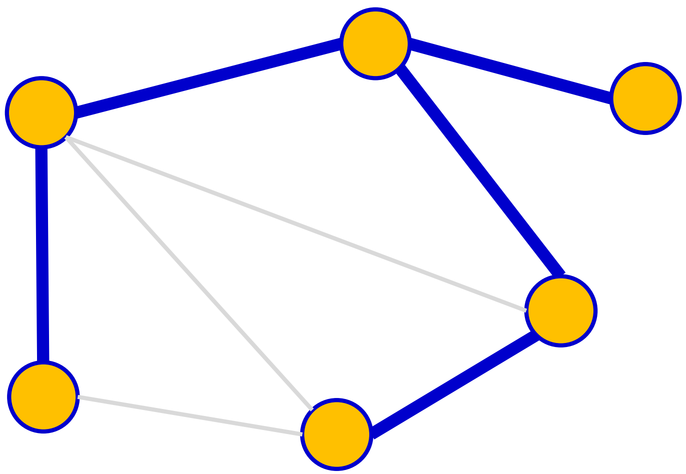
- Disegnare un grafo G che abbia le seguenti caratteristiche:
  a) G è connesso,
  b) la cardinalità del massimo abbinamento di G coincide con la cardinalità della minima copertura con nodi di G,
  c) la cardinalità del massimo insieme stabile di G sommata alla cardinalità della minima copertura con nodi di G è pari a 8

- Ottimizzazione combinatoria (OC)
- PLI per problemi di OC
- Il minimo albero ricoprente (MST)

Alberi di supporto

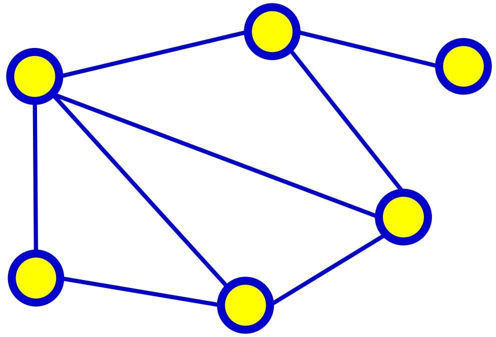
Sia $G = (V, E)$ un grafo simmetrico connesso.

Un albero di supporto (o albero ricoprente) $T = (V, F)$ di G è un grafo parziale di G aciclico e connesso.

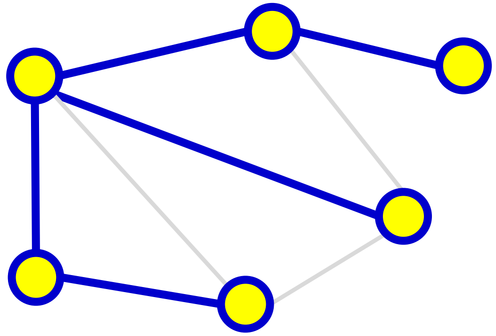


Alberi di supporto: proprietà di scambio

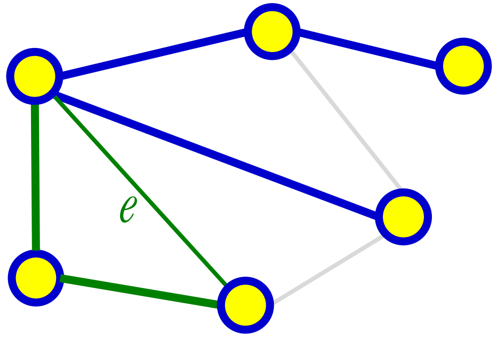
Sia $G = (V, E)$ un grafo simmetrico connesso



e $T = (V, F)$ un albero di supporto di G

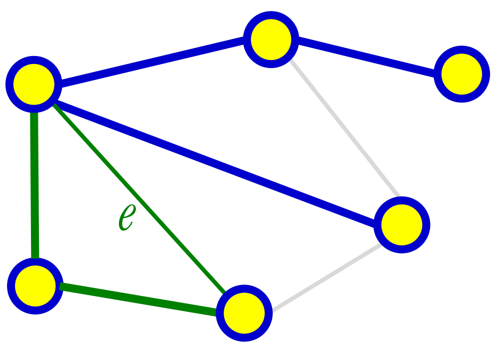


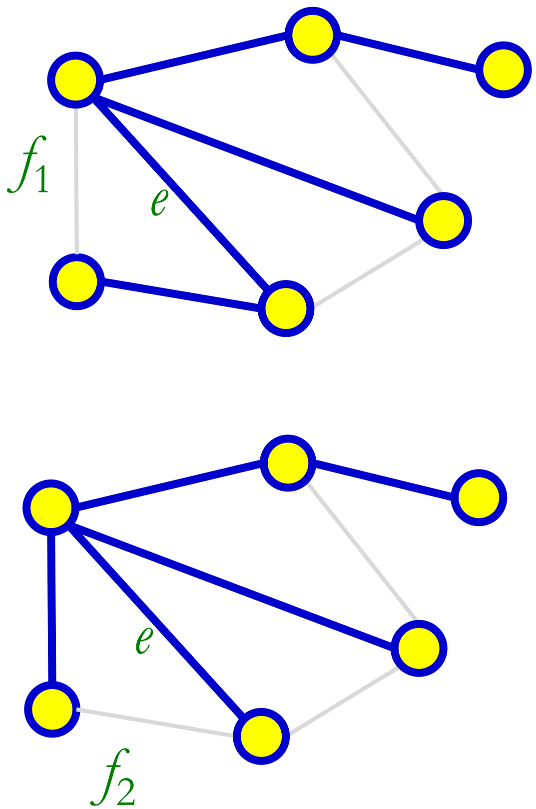
Sia $e \in E \setminus F$ e, in base alla proprietà degli alberi, C l'unico ciclo contenuto in $F \cup \{e\}$



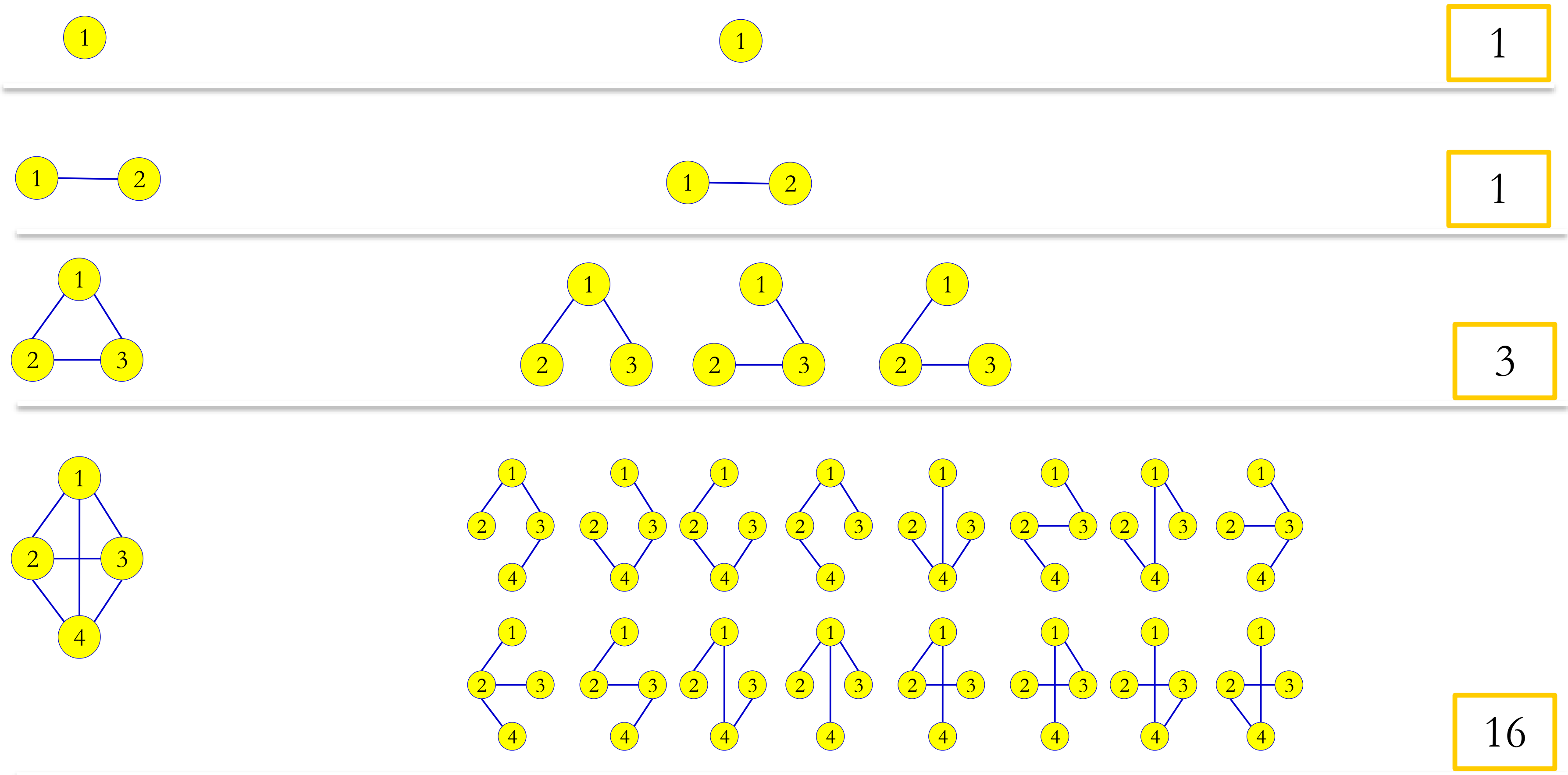
Alberi di supporto: proprietà di scambio

Proprietà di scambio: Per ogni arco $f \in C \setminus \{e\}$, $T' = (V, F \cup \{e\} \setminus \{f\})$ è un albero di supporto di G.





Alberi di supporto (etichettati): quanti sono?



Quanti alberi di supporto etichettati ha K_5 ?

Il numero di alberi ricoprenti

Il numero T_n di alberi ricoprenti è <u>al più</u> pari al numero di modi per scegliere $n-1$ archi di K_n

$$\binom{\binom{n}{2}}{n-1} \leq \binom{n^2}{n-1} =$$

$$\frac{n^2(n^2-1)\ldots(n^2-n+2)}{(n-1)!} =$$

$$\underbrace{\frac{n^2}{n-1} \cdot \frac{n^2-1}{n-2} \cdot \ldots \cdot \frac{n^2-n+2}{1}}_{n-1}$$

Il numero T_n di alberi ricoprenti è al più pari al numero di modi per scegliere $n-1$ archi di K_n

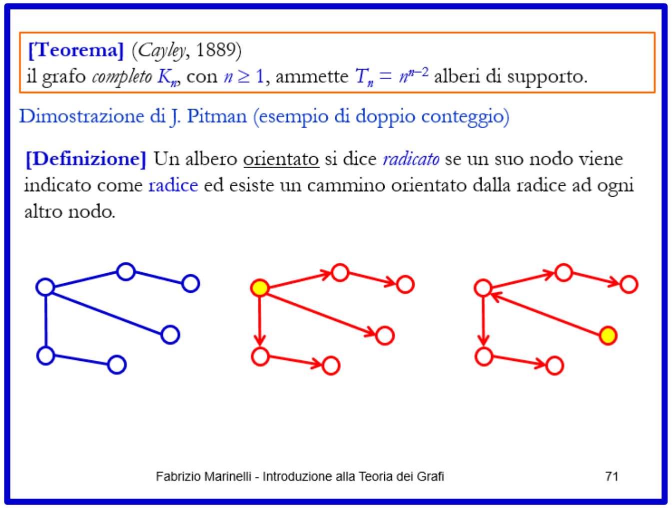
siccome $$\frac{n^2-k}{n-k-1} < \frac{n^2-k-1}{n-k-2}$$ per ogni $k < n-1$

possiamo scrivere

$$\binom{n^2}{n-1} \leq \underbrace{\frac{n^2}{n-1} \cdot \frac{n^2-1}{n-2} \cdot \ldots \cdot \frac{n^2-n+2}{1}}_{n-1} \leq \boxed{(n^2-n+2)^{n-1}}$$

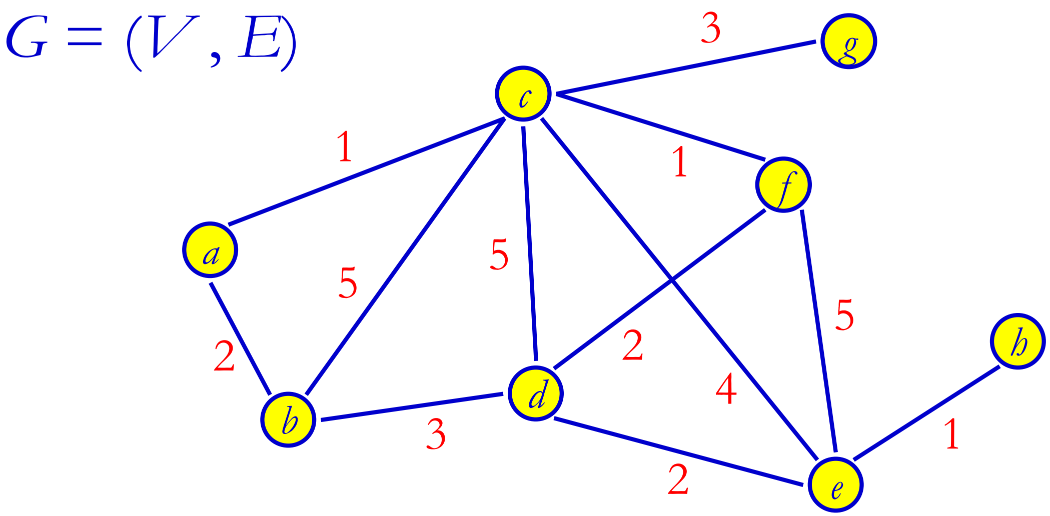
[Teorema] (*Cayley*, 1889)
il grafo *completo* K_n, con $n \geq 1$, ammette $T_n = n^{n-2}$ alberi di supporto.

Dimostrazione di J. Pitman (esempio di doppio conteggio)



Minimo Albero Ricoprente (MST)

Sia $G = (V, E)$ un grafo *pesato* sugli archi, cioè sia $c: E \to \mathbb{R}$ una funzione che associa un numero reale ad ogni arco di G

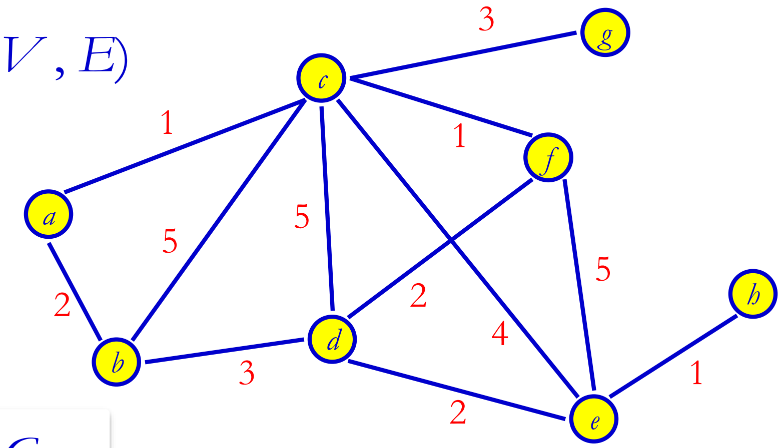


Il costo di un albero ricoprente $T = (V, F)$ del grafo G è dato dalla somma dei pesi degli archi in F:

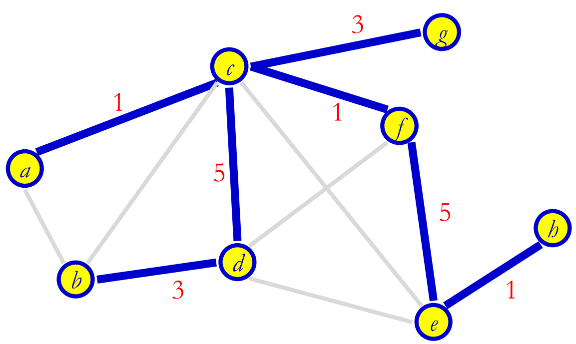
$$\boxed{c(T) = \sum_{e \in F} c_e}$$

Minimo Albero Ricoprente (MST)

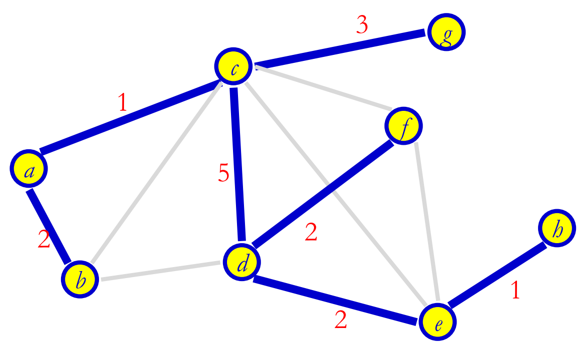
$G = (V, E)$



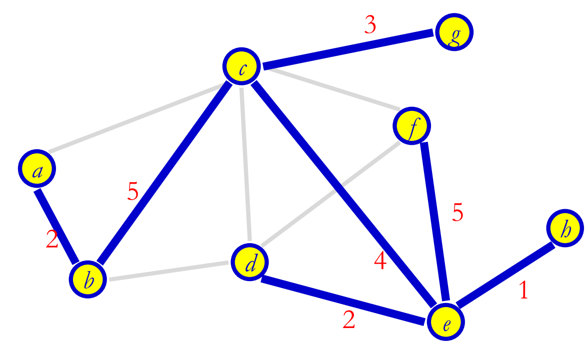
alcuni alberi ricoprenti di G



$T_1 = (V, F_1)$ $\quad c(T_1) = 19$



$T_2 = (V, F_2)$ $\quad c(T_2) = 16$



$T_3 = (V, F_3)$ $\quad c(T_3) = 22$

Qual è un albero ricoprente di peso minimo?

Applicazione

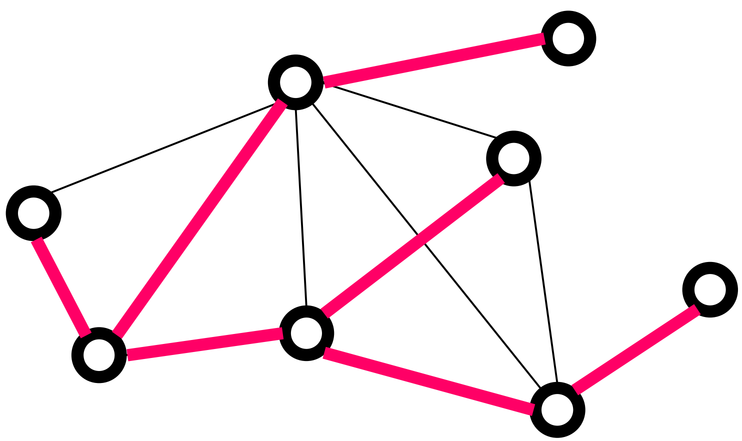
[Problema] Un'organizzazione segreta deve decidere come ogni affiliato possa comunicare un messaggio in *broadcasting* a tutti gli altri nel modo più sicuro possibile.

Definiamo un grafo $G = (V, E)$ in cui i nodi sono gli affiliati e esiste un arco $\{u, v\}$ se gli affiliati u e v possono comunicare <u>direttamente</u>. Il costo p_{uv} dell'arco indica la probabilità di intercettazione durante lo scambio di informazioni tra u e v.

Requisiti:

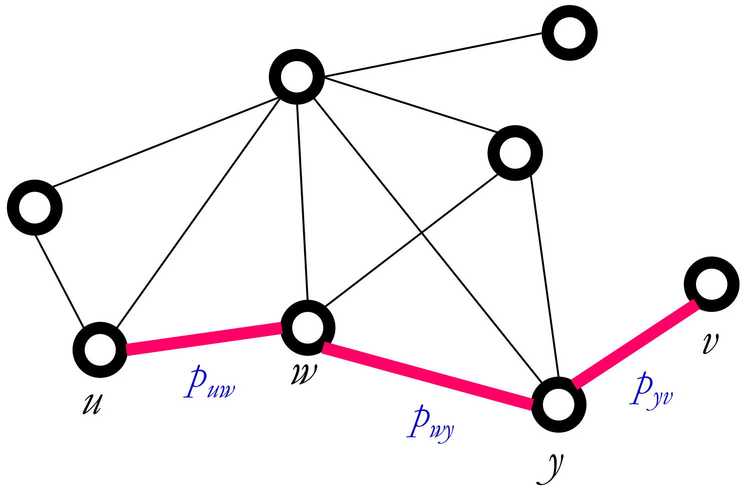
- ogni affiliato deve essere raggiungibile (**grafo connesso**)
- occorre evitare comunicazioni ridondanti (**grafo aciclico**)

Albero ricoprente T



Applicazione

Sia Q un cammino che collega u con v



$prob$(msg intercettato in Q)

$$\equiv$$

$1 - prob$(msg **non** intercettato in Q)

$$= 1 - \prod_{(i,j)\in Q} (1 - p_{ij})$$

… estendo il ragionamento a T

(trasmissione in broadcasting)

Le soluzioni ottime non cambiano se utilizzo una funzione monotona crescente in funzione obiettivo, come per esempio il logaritmo

$$\max \log \prod_{(i,j)\in T} (1 - p_{ij}) = \max \sum_{(i,j)\in T} \log(1 - p_{ij}) = \min \sum_{(i,j)\in T} -\log(1 - p_{ij})$$

Modello di PLI

$T \subseteq E$ è un albero ricoprente di G se il sottografo indotto è aciclico e connesso

Qual è l'albero ricoprente di G di peso minimo?

Modello di PLI …

$T \subseteq E$ è un albero ricoprente di G se il sottografo indotto è aciclico e connesso

Qual è l'albero ricoprente di G di peso minimo?

$$x_{uv} = \begin{cases} 1 \text{ se } \{u, v\} \in T \\ 0 \text{ altrimenti} \end{cases}$$

MST

$$\min \sum_{\{u,v\} \in E} c_{uv} x_{uv}$$

insieme di archi di peso minimo

$$\sum_{u \in S\; v \notin S} x_{uv} \geq 1 \qquad \forall\, \emptyset \subset S \subset V$$

vincolo di connessione.

$$0 \leq x_{uv} \leq 1, intero \qquad \forall \{u, v\} \in E$$

L'aciclicità è implicata dalla f.o. e dai pesi ≥ 0

Algoritmo di Kruskal (1956)

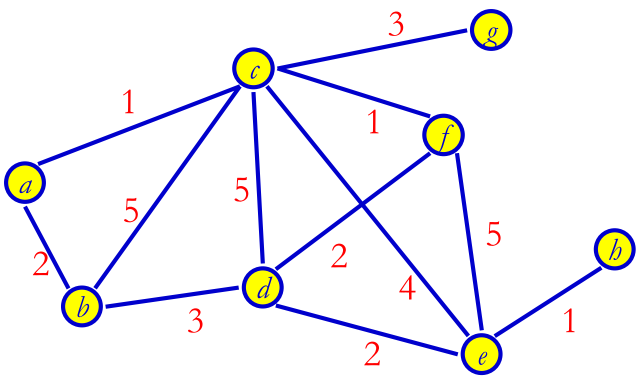
$$G = (V, E) \qquad |V| = n,\ |E| = m$$

[Idea] selezione degli archi con un criterio Greedy (cioè a partire da quello che pesa meno)

Algoritmo Kruskal

1. **Inizializzazione** $T = \emptyset$; ordina E per pesi non decrescenti
2. Se $|T| = n - 1$ allora **fine.**
3. altrimenti scegli l'arco $e \in E$ di peso minimo e poni $E = E \setminus \{e\}$
4. Se $T \cup \{e\}$ non forma cicli allora poni $T = T \cup \{e\}$
5. Vai al punto **2.**

Algoritmo di Kruskal: esempio



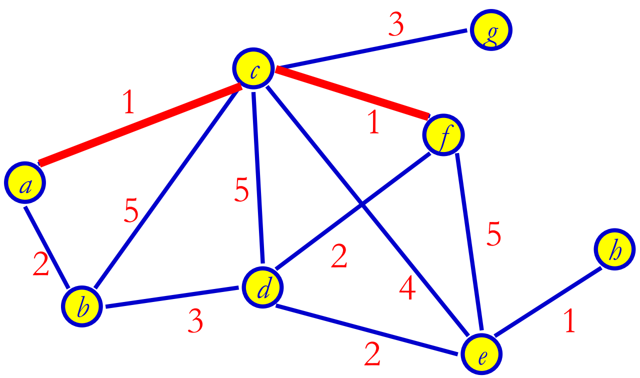
Ordina gli archi per pesi non decrescenti

steps

E

$\{a,c\}$	1
$\{c,f\}$	1
$\{e,h\}$	1
$\{a,b\}$	2
$\{d,e\}$	2
$\{d,f\}$	2
$\{b,d\}$	3
$\{c,g\}$	3
$\{c,e\}$	4
$\{e,f\}$	5
$\{c,h\}$	5
$\{c,d\}$	5

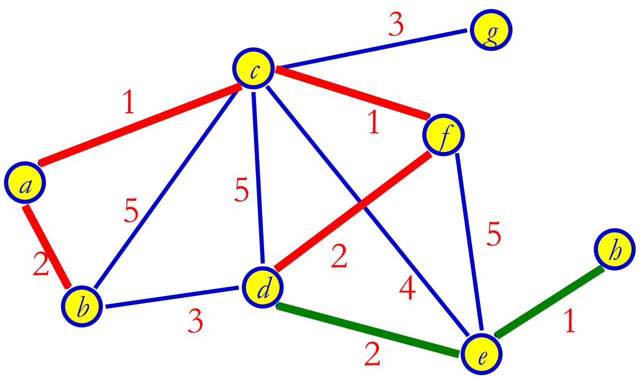
Algoritmo di Kruskal: esempio



Seleziona il prossimo arco (che non forma cicli con gli archi già selezionati)

			1	2	steps
E	$\{a,c\}$	1	$\{a,c\}$ 1		
	$\{c,f\}$	1		$\{c,f\}$ 1	
	$\{e,h\}$	1			
	$\{a,b\}$	2			
	$\{d,e\}$	2			
	$\{d,f\}$	2			
	$\{b,d\}$	3			
	$\{c,g\}$	3			
	$\{c,e\}$	4			
	$\{e,f\}$	5			
	$\{c,b\}$	5			
	$\{c,d\}$	5			

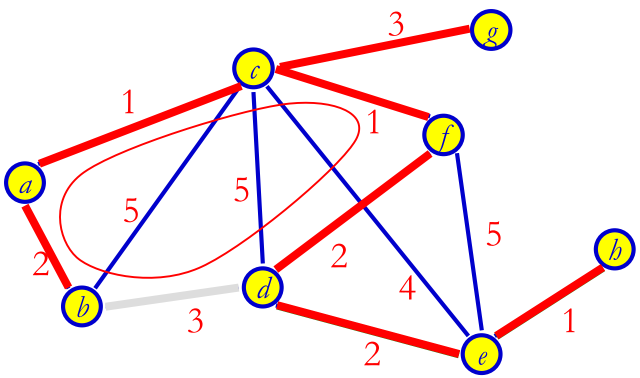
Algoritmo di Kruskal: esempio



Gli archi selezionati possono indurre più componenti connesse

			1	2	3	4	5	6	steps
	$\{a,c\}$	1	$\{a,c\}$ 1						
	$\{c,f\}$	1		$\{c,f\}$ 1					
	$\{e,h\}$	1			$\{e,h\}$ 1				
	$\{a,b\}$	2				$\{a,b\}$ 2			
	$\{d,e\}$	2					$\{d,e\}$ 2		
E	$\{d,f\}$	2						$\{d,f\}$ 2	
	$\{b,d\}$	3							
	$\{c,g\}$	3							
	$\{c,e\}$	4							
	$\{e,f\}$	5							
	$\{c,b\}$	5							
	$\{c,d\}$	5							

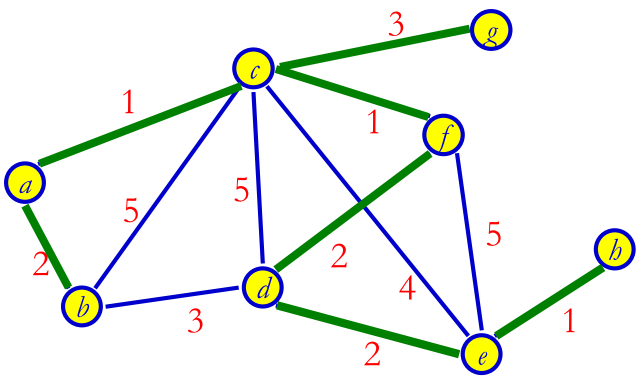
Algoritmo di Kruskal: esempio



Se l’arco forma un ciclo, viene escluso

E		1	2	3	4	5	6	7	8	steps
$\{a,c\}$	1	$\{a,c\}$ 1								
$\{c,f\}$	1		$\{c,f\}$ 1							
$\{e,h\}$	1			$\{e,h\}$ 1						
$\{a,b\}$	2				$\{a,b\}$ 2					
$\{d,e\}$	2					$\{d,e\}$ 2				
$\{d,f\}$	2						$\{d,f\}$ 2			
$\{b,d\}$	3							$\{b,d\}$		
$\{c,g\}$	3								$\{c,g\}$ 3	
$\{c,e\}$	4									
$\{e,f\}$	5									
$\{c,b\}$	5									
$\{c,d\}$	5									

Algoritmo di Kruskal: esempio



sono stati selezionati $n - 1$ archi: **STOP**

$$\boxed{c(T) = 12}$$

E		1	2	3	4	5	6	7	8	steps
$\{a,c\}$	1	$\{a,c\}$ 1								
$\{c,f\}$	1		$\{c,f\}$ 1							
$\{e,h\}$	1			$\{e,h\}$ 1						
$\{a,b\}$	2				$\{a,b\}$ 2					
$\{d,e\}$	2					$\{d,e\}$ 2				
$\{d,f\}$	2						$\{d,f\}$ 2			
$\{b,d\}$	3									
$\{c,g\}$	3								$\{c,g\}$ 3	
$\{c,e\}$	4									
$\{e,f\}$	5									
$\{c,b\}$	5									
$\{c,d\}$	5									

Algoritmo di Kruskal: costo computazionale

Algoritmo Kruskal

1. Inizializzazione

2. Selezione di $n-1$ archi

$O(n^2 \log n)$

Inizializzazione: ordinamento dei pesi degli m archi.

Ordinare m elementi costa $O(m \log m) = O(n^2 \log n)$ dato che $m = O(n^2)$

Algoritmo di Kruskal: costo computazionale

Algoritmo Kruskal	
1. Inizializzazione	$O(n^2 \log n)$
2. Selezione di $n - 1$ archi	$O(n^2)$

Selezione archi: $O(n)$ archi selezionati e ogni selezione ha un costo computazionale di $O(n)$ se si usa la seguente struttura per verificare che non si formino cicli:

$$C = \underbrace{\boxed{}\boxed{} \quad \cdots \quad \boxed{}}_{n \text{ elementi}}$$

$C[i]$ = indice della componente connessa che contiene il nodo i (inizialmente 0)

- L'arco selezionato $\{i, j\}$ forma un ciclo se $C[i] = C[j]$ (un solo confronto).
- Se l'arco selezionato $\{i, j\}$ unisce 2 componenti connesse, si aggiornano nel caso peggiore (ultima iterazione) n etichette.

Algoritmo di Kruskal: costo computazionale

Algoritmo Kruskal

1. **Inizializzazione** $O(n^2 \log n)$ $+$
2. **Selezione** di $n - 1$ archi $O(n^2)$

il costo totale è quindi $O(n^2 \log n + n^2) = \boxed{O(n^2 \log n)}$

Algoritmo di Prim

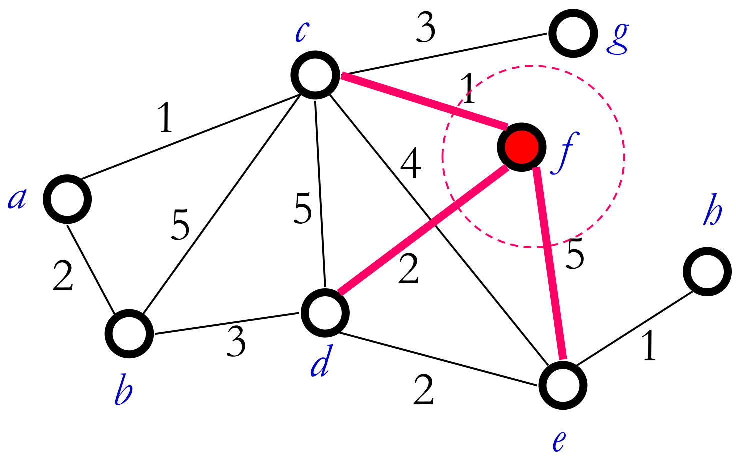
$$G = (V, E) \qquad |V| = n,\ |E| = m$$

[Idea] Si costruisce l'albero ricoprente T partendo da un nodo <u>qualsiasi</u> e aggiungendo di volta il volta l'arco di minor peso <u>adiacente</u> all'albero parziale corrente.

Algoritmo Prim

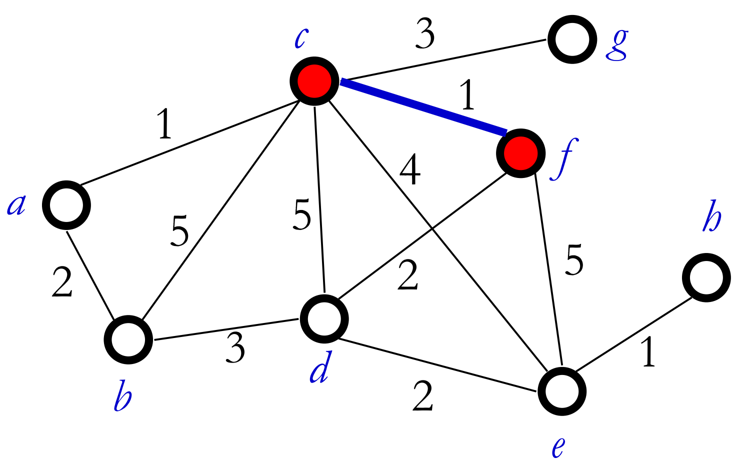
1. **Inizializzazione** $T = \emptyset$; $S = \{u\}$, $u \in V$
2. Se $|T| = n - 1$ allora **fine**.
3. altrimenti scegli $(u,v) \in E$ di peso minimo con $u \in S$ e $v \in V \setminus S$
4. $T \cup \{(u,v)\}$, $S = S \cup \{v\}$
5. Vai al punto **2**.

Algoritmo di Prim: esempio



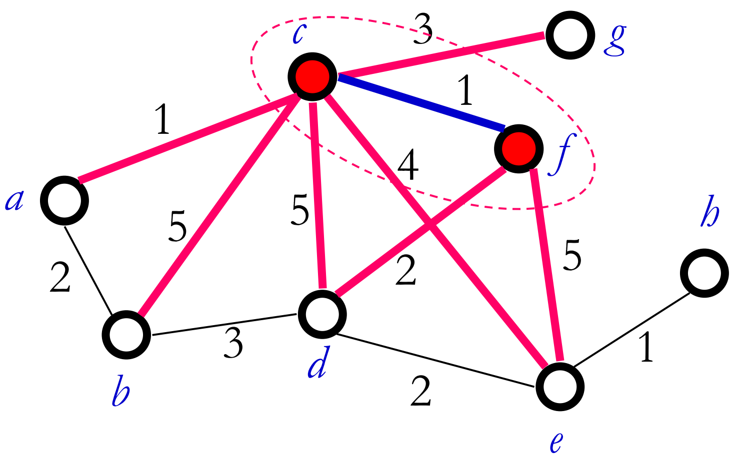
- Inizializzo S con il nodo f e valuto tutti gli archi con un estremo in S e l'altro in $V \setminus S$ (archi in rosso)

$c(T) = 1$

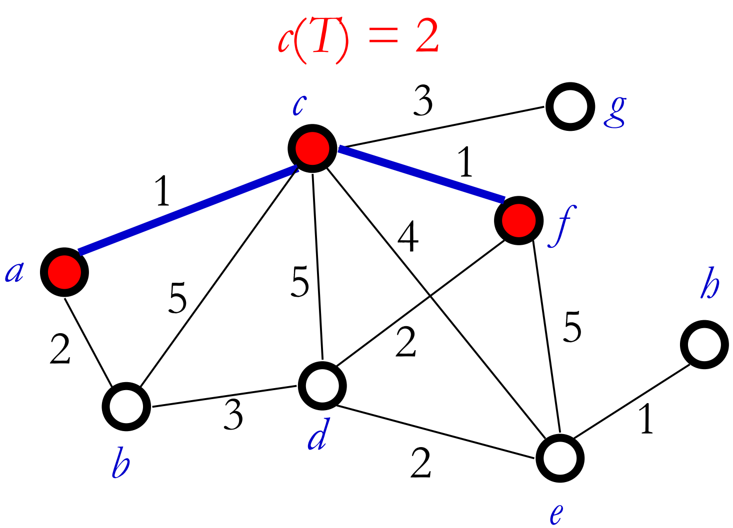


- Tra gli archi valutati, seleziono quello di costo minimo (arco blu) e lo aggiungo all'albero corrente. Aggiungo all'insieme S il nodo c

Algoritmo di Prim: esempio

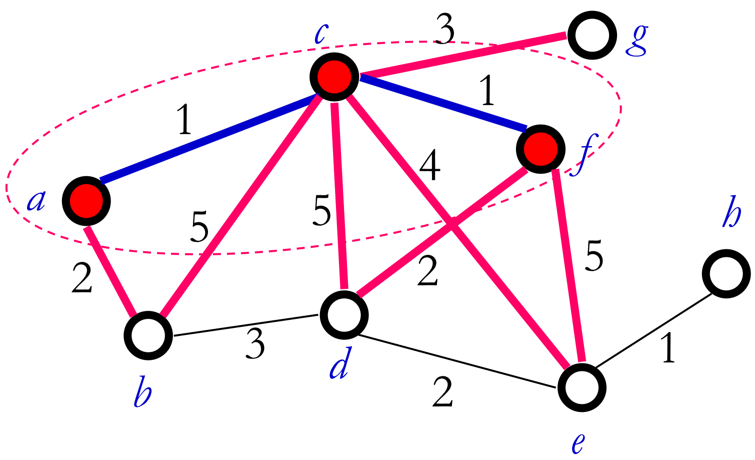


- Valuto tutti gli archi con un estremo in S e l'altro in $V \setminus S$ (archi in rosso)

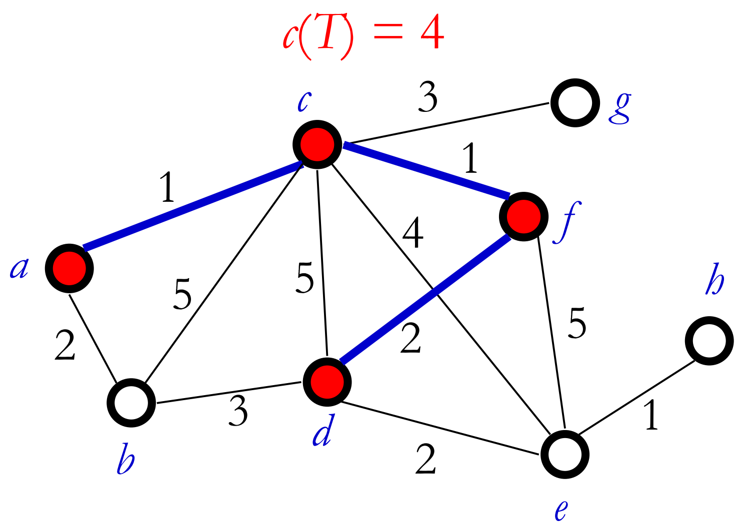


- Tra gli archi valutati, seleziono quello di costo minimo (arco blu) e lo aggiungo all'albero corrente. Aggiungo all'insieme S il nodo a

Algoritmo di Prim: esempio

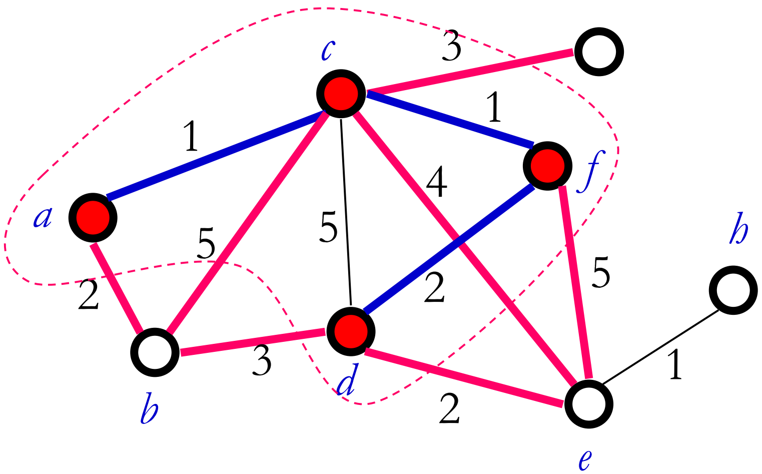


▶ Valuto tutti gli archi con un estremo in S e l'altro in $V \setminus S$ (archi in rosso)



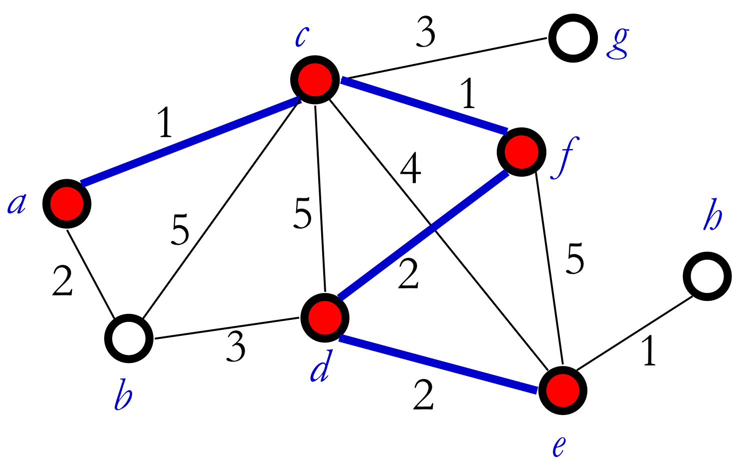
▶ Tra gli archi valutati, seleziono quello di costo minimo (arco blu) e lo aggiungo all'albero corrente. Aggiungo all'insieme S il nodo d

Algoritmo di Prim: esempio



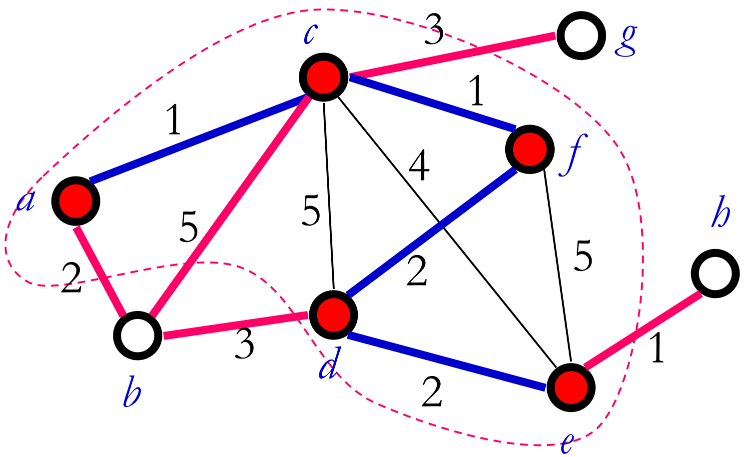
- Valuto tutti gli archi con un estremo in S e l'altro in $V \setminus S$ (archi in rosso)

$c(T) = 6$



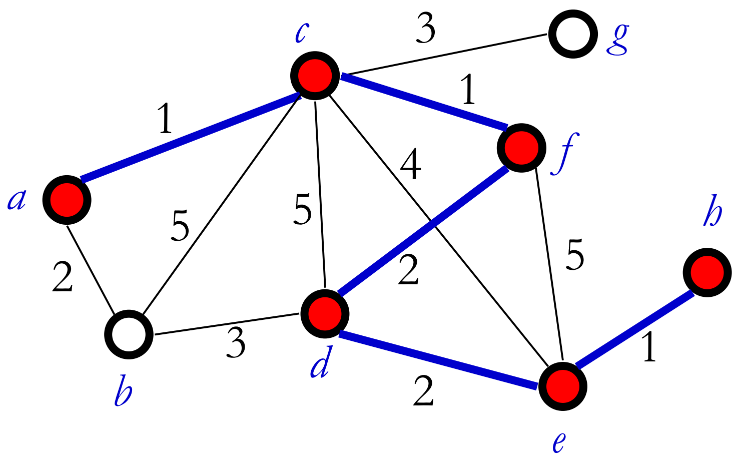
- Tra gli archi valutati, seleziono quello di costo minimo (arco blu) e lo aggiungo all'albero corrente. Aggiungo all'insieme S il nodo e

Algoritmo di Prim: esempio



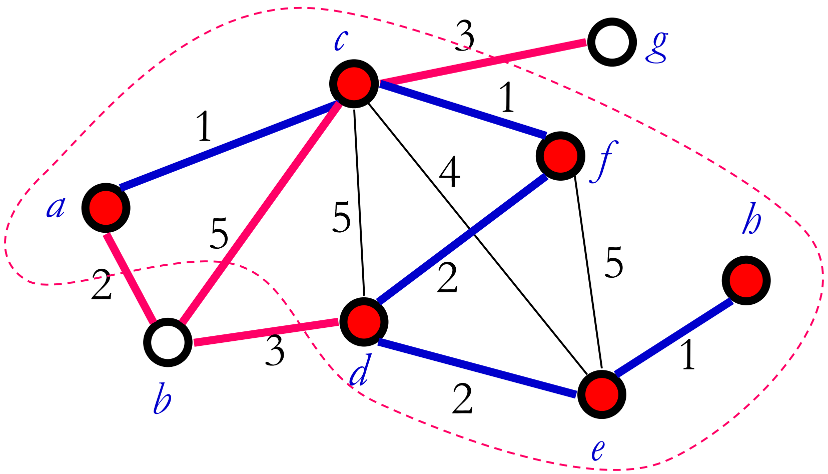
- Valuto tutti gli archi con un estremo in S e l'altro in $V \setminus S$ (archi in rosso)

$c(T) = 7$



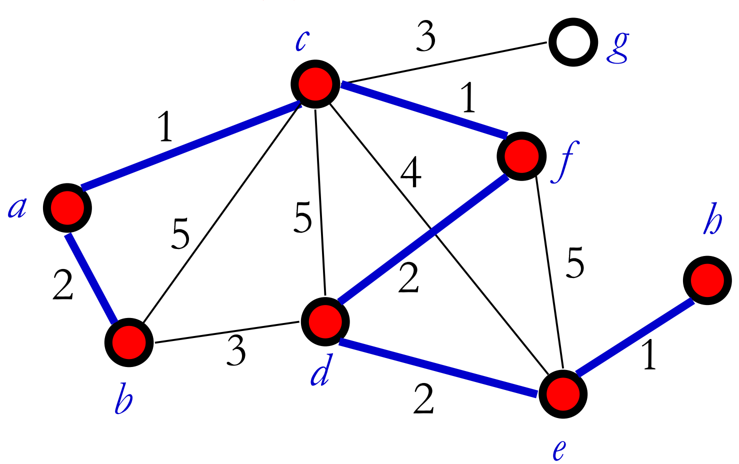
- Tra gli archi valutati, seleziono quello di costo minimo (arco blu) e lo aggiungo all'albero corrente. Aggiungo all'insieme S il nodo h

Algoritmo di Prim: esempio



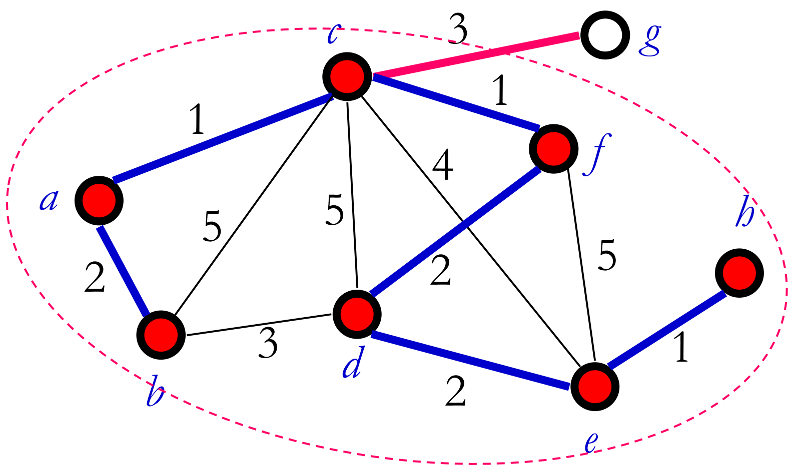
▶ Valuto tutti gli archi con un estremo in S e l'altro in $V \setminus S$ (archi in rosso)

$c(T) = 9$

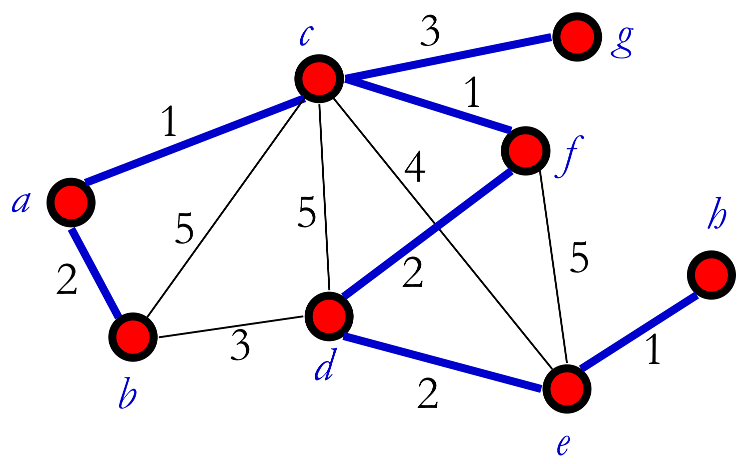


▶ Tra gli archi valutati, seleziono quello di costo minimo (arco blu) e lo aggiungo all'albero corrente. Aggiungo all'insieme S il nodo b

Algoritmo di Prim: esempio



$c(T) = 12$



Algoritmo di Prim: costo computazionale

Algoritmo Prim

1. **Inizializzazione** $O(1)$ +
2. **Selezione** di $n - 1$ archi $O(n^2)$

$\boxed{O(n^2)}$

Selezione archi: $O(n)$ archi selezionati e ogni selezione richiede l'ispezione di al più n archi

Con liste di adiacenza e heap di Fibonacci il costo totale può essere ridotto a $O(n\log n)$

Prim e Kruskal: esattezza

- Gli algoritmi di Kruskal e Prim sono algoritmi *greedy*.
- La scelta *greedy* consiste nella selezione di un arco di peso minimo (che non forma cicli nel sottografo aciclico corrente).

[Teorema] Gli algoritmi di Kruskal e Prim sono **algoritmi esatti**, cioè forniscono <u>sempre</u> una soluzione ottima (un albero ricoprente di peso minimo).

Gli algoritmi *greedy* sono esatti **<u>solo</u>** per problemi di ottimizzazione che hanno una **<u>struttura particolare</u>**

Esercizi

- Descrivere un algoritmo *greedy* per il problema (versione pesata) di
  1. massimo insieme stabile,
  2. massima clique,
  3. assegnamento massimo,
  4. minima copertura con archi,
  5. minima copertura con nodi.

- Verificare che nei casi precedenti un algoritmo *greedy* fornisce sempre un insieme massimale (minimale) ma non sempre un insieme massimo (minimo).

Discussione

- **sfatiamo alcuni miti**

  - «Se il modello di PLI non è *compatto* (non è cioè di dimensione polinomiale) il problema è difficile».

    Il modello ha m variabili e un 2^n vincoli, uno per ogni possibile sottoinsieme di nodi

  - «Se le soluzioni ammissibili sono molte trovare quella ottima è più difficile»

    I possibili alberi ricoprenti sono n^{n-2} ma esiste almeno un algoritmo polinomiale esatto.

Letture ricreative e contenuti multimediali

1. P. M. Higgins,
   La matematica dei social network. Una introduzione alla teoria dei grafi
   Dedalo, 2011

2. Brian Eno
   Thursday Afternoon
   1985

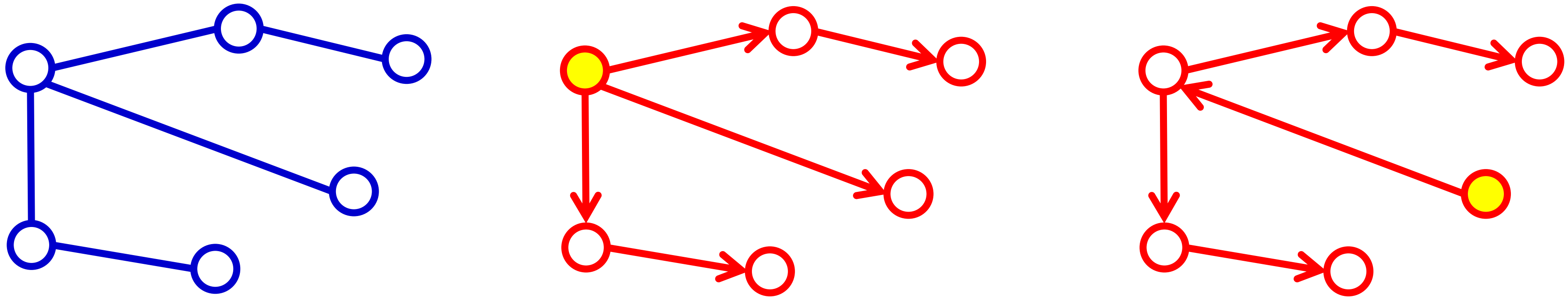
APPENDICE:

dimostrazione del teorema di Cayley

[Teorema] (*Cayley*, 1889)
il grafo *completo* K_n, con $n \geq 1$, ammette $T_n = n^{n-2}$ alberi di supporto.

Dimostrazione di J. Pitman (esempio di doppio conteggio)

[Definizione] Un albero orientato si dice *radicato* se un suo nodo viene indicato come radice ed esiste un cammino orientato dalla radice ad ogni altro nodo.



[Teorema] (*Cayley*, 1889)
il grafo *completo* K_n, con $n \geq 1$, ammette $T_n = n^{n-2}$ alberi di supporto.

Dimostrazione di J. Pitman (esempio di doppio conteggio)

τ = numero di modi (cioè le possibili sequenze di scelta di archi) in cui si può costruire un albero *radicato* a partire da un grafo vuoto con n nodi

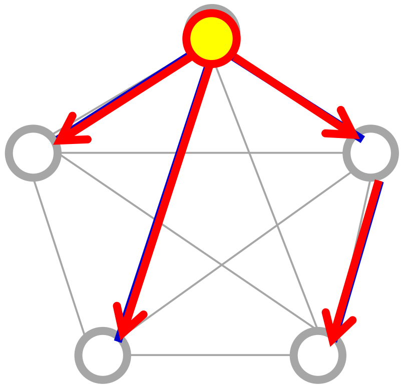
Il valore τ può essere ottenuto in due modi:

1. considerando i possibili alberi ricoprenti (non radicati), oppure
2. direttamente

[Teorema] (*Cayley*, 1889)
il grafo *completo* K_n, con $n \geq 1$, ammette $T_n = n^{n-2}$ alberi di supporto.

Dimostrazione di J. Pitman (esempio di doppio conteggio)

1. Si consideri un qualsiasi albero ricoprente T di K_n e si scelga un nodo di T come radice



(si osservi che così facendo la direzione di ogni arco è totalmente determinata).

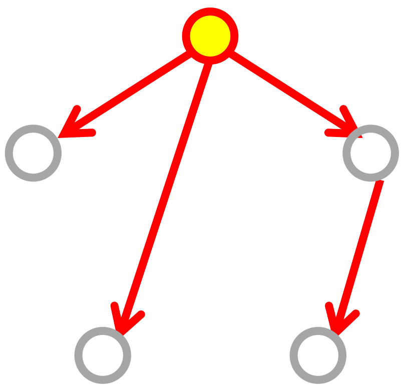
[Teorema] (*Cayley*, 1889)
il grafo *completo* K_n, con $n \geq 1$, ammette $T_n = n^{n-2}$ alberi di supporto.

Dimostrazione di J. Pitman (esempio di doppio conteggio)

1. Gli archi di T sono $(n - 1)$ quindi le possibili sequenze di scelta sono $(n - 1)!$

   I possibili nodi radice sono n, quindi in totale ci sono $n\,(n - 1)!$ modi per costruire un albero radicato a partire da T.

   Siccome gli alberi ricoprenti di K_n sono T_n si ha:



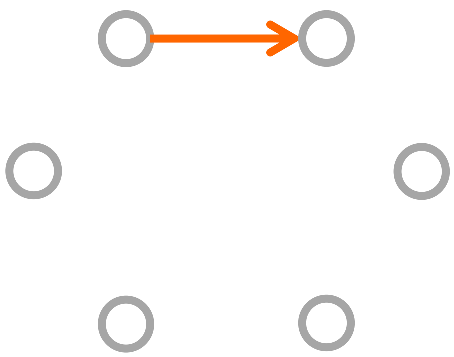
$$\boxed{\tau = T_n\, n\,(n - 1)! = T_n\, n!}$$

[Teorema] (*Cayley*, 1889)
il grafo *completo* K_n, con $n \geq 1$, ammette $T_n = n^{n-2}$ alberi di supporto.

Dimostrazione di J. Pitman (esempio di doppio conteggio)

2. Prendi un grafo vuoto con n nodi e costruisci un albero radicato aggiungendo un arco orientato alla volta.

   - Al primo passo ci sono $n_1 = n(n-1)$ modi possibili: ognuno degli n nodi può essere scelto come coda dell'arco e solo $(n-1)$ nodi come testa.

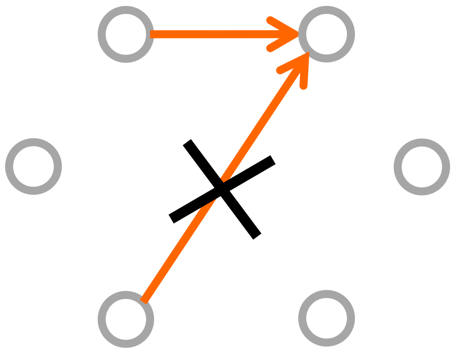


[Teorema] (*Cayley*, 1889)
il grafo *completo* K_n, con $n \geq 1$, ammette $T_n = n^{n-2}$ alberi di supporto.

Dimostrazione di J. Pitman (esempio di doppio conteggio)

2. Prendi un grafo vuoto con n nodi e costruisci un albero radicato aggiungendo un arco orientato alla volta.

   - al 2° passo ci sono $n_2 = n(n-2)$ modi possibili: di nuovo ognuno degli n nodi può essere scelto come coda dell'arco, ma ora solo $(n-2)$ nodi possono essere la testa.

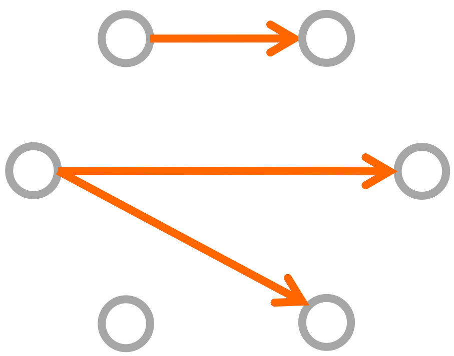


Osservazione chiave: ogni nodo ha al più un arco "entrante" (0 la radice e 1 tutti gli altri nodi)

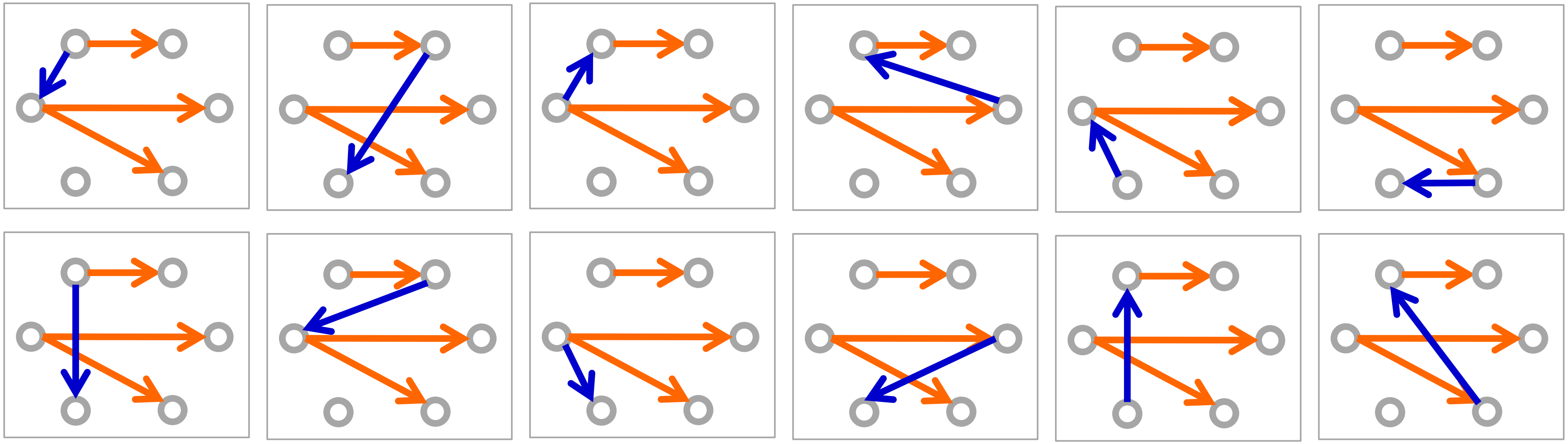
[Teorema] (*Cayley*, 1889)
il grafo *completo* K_n, con $n \geq 1$, ammette $T_n = n^{n-2}$ alberi di supporto.

Dimostrazione di J. Pitman (esempio di doppio conteggio)

2. Prendi un grafo vuoto con n nodi e costruisci un albero radicato aggiungendo un arco orientato alla volta.
   - al k-esimo passo, $k - 1$ nodi sono già stati scelti come testa e quindi il nuovo arco può avere ognuno degli n nodi come coda ma solo $n - k$ nodi come testa.



$$n_4 = 6(6 - 4)$$



$$n_k = n(n-k)$$

[Teorema] (*Cayley*, 1889)
il grafo *completo* K_n, con $n \geq 1$, ammette $T_n = n^{n-2}$ alberi di supporto.

Dimostrazione di J. Pitman (esempio di doppio conteggio)

2. Prendi un grafo vuoto con n nodi e costruisci un albero radicato aggiungendo un arco orientato alla volta.
   - In generale, le possibili scelte al passo k-esimo sono $n_k = n(n-k)$. Quindi

$$\tau = \prod_{k=1}^{n-1} n_k = n^{n-1}(n-1)! = n^{n-2} n!$$

[Teorema] (*Cayley*, 1889)
il grafo *completo* K_n, con $n \geq 1$, ammette $T_n = n^{n-2}$ alberi di supporto.

Dimostrazione di J. Pitman (esempio di doppio conteggio)

$$\tau = T_n n!$$

$$\tau = n^{n-2} n!$$

$$T_n n! = n^{n-2} n!$$

$$\boxed{T_n = n^{n-2}}$$

V.L.A.D.