

Ottimizzazione Combinatoria

A. A. 2012-2013

Docente: Mara Servilio

Orario delle lezioni: Martedì 11:15-13:15; Mercoledì 11:15-13:15

E-mail: mara.servilio@di.univaq.it

Sito web: <http://www.di.univaq.it/~oil>

Propedeuticità

- Studenti immatricolati con la legge 509: **Ricerca Operativa, Algoritmi e Strutture Dati**
- Studenti immatricolati con la legge 270: No propedeuticità

Introduzione

Problema di Ottimizzazione

Siano

- U = insieme universo, ossia un insieme di soluzioni, decisioni e alternative.
- $F \subseteq U$ = insieme ammissibile definito tramite una serie di relazioni dette vincoli.
- $f: U \rightarrow \mathbf{R}$ = funzione obiettivo.
- Direzione di ottimizzazione: minimo o massimo.

Problema di ottimizzazione (in forma di minimo)

Trovare un elemento $x^* \in F$ tale che $f(x^*) \leq f(x)$ per ogni $x \in F$.

$f(x^*)$ = valore ottimo

x^* = soluzione ottima

Esempio

Un'azienda di spedizioni, proprietaria di alcuni treni merci, intende realizzare un servizio di spedizioni tra due diverse località via ferrovia.

L'azienda

1. Chiede gli orari disponibili alla società che gestisce la rete ferroviaria e i costi relativi.

U = insieme di tutti i possibili assegnamenti orario-servizio.

$F \subseteq U$ insieme degli assegnamenti che rispettano alcuni vincoli fisici.

$f : U \rightarrow \mathbf{R}$ funzione che esprime, per ogni scelta possibile, il guadagno ottenuto a fronte del costo.

2. Cerca l'assegnamento $x^* \in F$ che massimizza il guadagno.

Esempio

L'azienda deve risolvere un problema di ottimizzazione.

Istanza:

- O insieme dei possibili orari, ognuno con il proprio costo.
- T insieme dei treni merci disponibili, ognuno con il proprio profitto (ogni treno è associato ad un servizio).

Problema: Assegnare un sottoinsieme $T' \subseteq T$ di treni merci a un sottoinsieme $O' \subseteq O$ di orari in modo da

- massimizzare il guadagno
- rispettare i vincoli fisici.

Esempio

Il gestore della rete ferroviaria

1. Studia la fattibilità delle richieste della società.
2. Pianifica alcuni orari alternativi da proporre all'azienda.

U insieme degli orari alternativi possibili.

$F \subseteq U$ insieme degli orari che rispettano alcuni vincoli (standard di sicurezza).

3. Definisce i prezzi di ogni proposta.

$f: U \rightarrow \mathbf{R}$ funzione che esprime, per ogni proposta alternativa, il guadagno ottenuto dal gestore.

4. Cerca l'orario $x^* \in F$ (se esiste!) che massimizza il guadagno.

Esempio

Il gestore della rete deve risolvere un problema di ottimizzazione

Istanza:

- Intervallo di tempo necessario a percorrere ogni tratta della linea
- Intervallo di tempo minimo e massimo di sosta in ogni stazione
- Standard di sicurezza
- Orario esistente

Problema: Trovare (se esiste!) un orario che contenga il nuovo treno merci, che massimizzi il guadagno e che rispetti gli standard di sicurezza.

Problema di Ottimizzazione Combinatoria

Siano

- $N = \{1, 2, \dots, n\}$ insieme finito.
- c vettore di pesi (costi/profitti) con coordinate c_j definite per ogni $j \in N$
- Insieme universo: $U = \{\text{tutti i possibili } 2^{|N|} \text{ sottoinsiemi di } N\}$
- Famiglia ammissibile: $F = \{\text{sottoinsiemi } F \text{ di } U \text{ che soddisfano una certa proprietà } P\}$.

Problema di ottimizzazione combinatoria (in forma di minimo)

$$\min_{S \subseteq N} \left\{ \sum_{j \in S} c_j : S \in F \right\}$$

Alcune applicazioni

- Progetto di servizi logistici.
- Progetto di una rete di trasmissione radiotelevisiva.
- Pianificazione della produzione.
- ...

Problema dello zaino

Supponiamo di avere a disposizione un budget b per un insieme di possibili investimenti $I = \{1, \dots, n\}$ da effettuare nell'anno corrente.

Associamo ad ogni possibile investimento $i \in I$

- un costo $c_i > 0$,
- un guadagno $p_i > 0$.

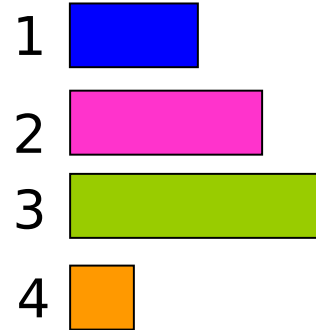
Problema: Scegliere il sottoinsieme di investimenti da effettuare in modo da massimizzare il guadagno e senza eccedere il budget a disposizione.

$U = \{\text{tutti i possibili } 2^n \text{ sottoinsiemi di } I\} = \text{insieme universo}$

$F = \{F \subseteq U : \text{il costo totale degli investimenti in } F \text{ non eccede il budget } b\} = \text{famiglia ammissibile}$

Problema dello zaino

Budget disponibile: $b = 5$



c	p
2	10
3	14
4	12
1	8

Insiemi ammissibili

\emptyset , {1}, {2}, {3}, {4}, {1,2}, {1,4}, {2,4}, {3,4}

Soluzione ottima

{1,2} di valore 24

Il numero di insiemi ammissibili è pari al numero di possibili sottoinsiemi di un insieme di n oggetti, ossia 2^n .

Problema dell'assegnamento

Consideriamo 3 artigiani e 3 lavori da realizzare.

I costi richiesti da ogni artigiano per ogni lavoro sono riportati nella seguente tabella

A \ L	1	2	3
1	10	12	20
2	7	15	18
3	14	10	9

Problema: Assegnare esattamente un artigiano ad un lavoro in modo da minimizzare i costi totali.

Problema dell'assegnamento

$U = \{\text{Possibili sottoinsiemi di coppie (artigiano, lavoro)}\} = \text{insieme universo}$

$F = \{F \subseteq U : \text{esiste esattamente una coppia per ogni artigiano}\} = \text{famiglia ammissibile}$

$A \setminus L$	1	2	3
1	10	12	20
2	7	15	18
3	14	10	9

Problema dell'assegnamento

Insiemi ammissibili

1. $\{(a_1, l_1); (a_2, l_2); (a_3, l_3)\}$ di costo 34
2. $\{(a_1, l_2); (a_2, l_3); (a_3, l_1)\}$ di costo 44
3. $\{(a_1, l_3); (a_2, l_1); (a_3, l_2)\}$ di costo 37
4. $\{(a_1, l_3); (a_2, l_2); (a_3, l_1)\}$ di costo 49
5. $\{(a_1, l_2); (a_2, l_1); (a_3, l_3)\}$ di costo 28
6. $\{(a_1, l_1); (a_2, l_3); (a_3, l_2)\}$ di costo 38

$A \setminus L$	1	2	3
1	10	12	20
2	7	15	18
3	14	10	9

La soluzione ottima ha valore 28.

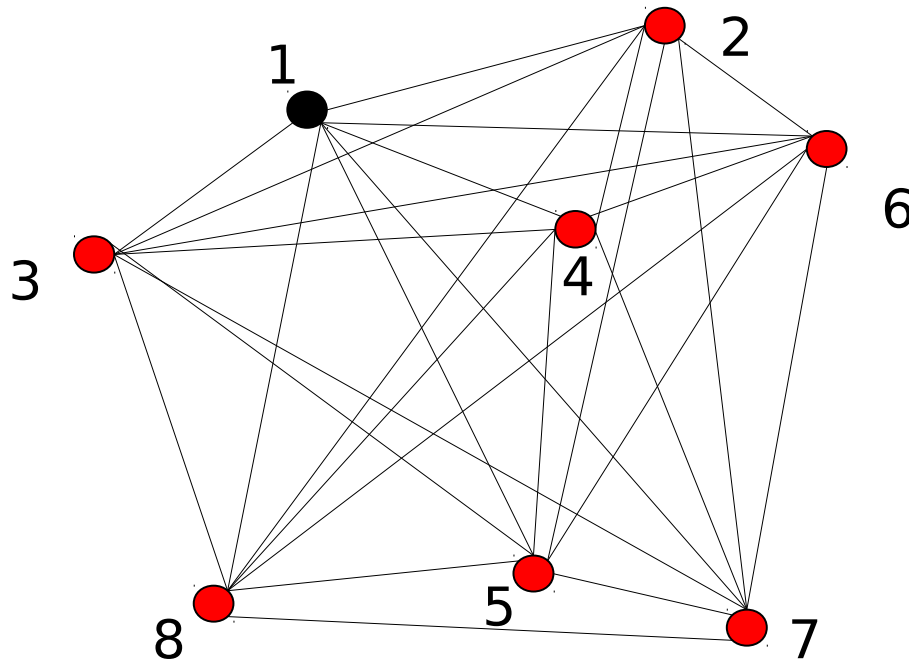
I possibili assegnamenti, e quindi i possibili insiemi ammissibili, sono 3!

In generale, su un insieme di n elementi esistono $n!$ insiemi ammissibili.

Problema del commesso viaggiatore (TSP)

Consideriamo n punti nel piano.

Per ogni coppia di punti (i, j) definiamo un costo $c_{ij} > 0$.

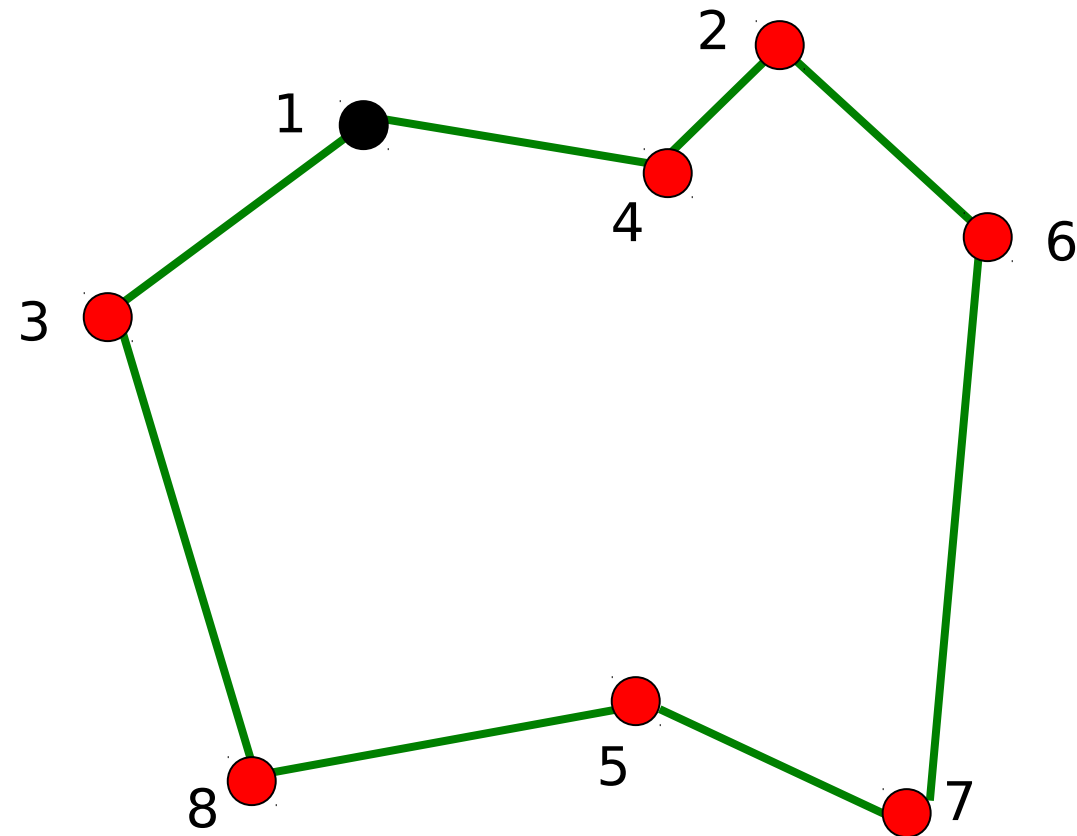


Problema: Determinare un percorso che inizi e termini con il nodo 1, che tocchi tutti i punti nel piano (*tour*) e il cui costo sia minimo.

Problema del commesso viaggiatore (TSP)

$U = \{\text{sottoinsiemi di tutte le possibili coppie } (i, j)\} = \text{insieme universo}$

$F = \{F \subseteq U : F \text{ individua un tour}\} = \text{famiglia ammissibile}$

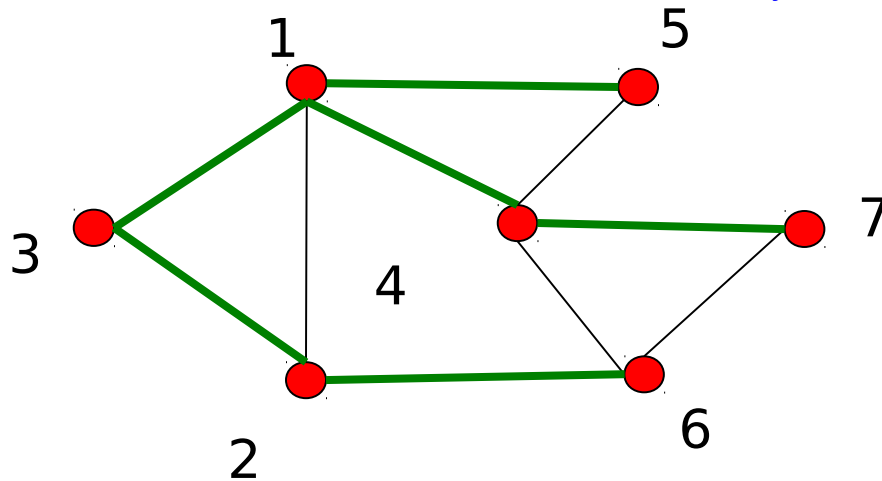


In un grafo completo con n nodi esistono $(n-1)!$ tour.

Problema del minimo albero ricoprente

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

Per ogni arco $(i, j) \in E$ definiamo un peso $w_{ij} \geq 0$.



Problema: Determinare un sottoinsieme di archi F che individui un sottografo connesso privo di cicli che contenga tutti i nodi di V (*albero ricoprente*) e il cui peso sia minimo.

Problema del minimo albero ricoprente

$U = \{\text{sottoinsiemi di archi del grafo } G\} = \text{insieme universo}$

$F = \{F \subseteq U : F \text{ individua un sottografo privo di cicli e contenente tutti i nodi di } G\} = \text{famiglia ammissibile}$

In un grafo completo con n nodi esistono n^{n-2} alberi ricoprenti.

Proprietà dei problemi di OC

1. Tutti i problemi di OC sono definiti su insiemi ammissibili finiti e numerabili.
2. Il valore della funzione obiettivo può essere calcolato in corrispondenza di ogni insieme ammissibile.

Esiste un algoritmo “universale” per i problemi di OC che si chiama enumerazione totale.

- Enumera tutti i possibili sottoinsiemi dell'insieme universo U
- Verifica se il sottoinsieme corrente F appartiene alla famiglia F
- Se F è ammissibile e la funzione obiettivo su F assume un valore migliore rispetto all'ottimo corrente, aggiorna la soluzione.

Proprietà dei problemi di OC

n	$\log n$	$n^{0.5}$	n^2	2^n	$n!$
10	3.32	3.16	100	1.02×10^3	3.6×10^6
100	6.64	10.00	10000	1.27×10^{30}	9.33×10^{157}
1000	9.97	31.62	1000000	1.07×10^{301}	4.02×10^{2567}

Considerando le prestazioni di un calcolatore moderno, l'algoritmo di enumerazione totale impiegherebbe circa 2 anni per risolvere un problema di commesso viaggiatore con 20 punti nel piano.

Obiettivi di questo corso

1. Studiare tecniche matematiche che consentano di progettare algoritmi “efficienti” per i problemi di OC:
 - Algoritmi ammissibili a complessità polinomiale (Assegnamento).
 - Algoritmi ammissibili a complessità pseudo-polinomiale (Zaino).
 - Algoritmi ammissibili a complessità non polinomiale (TSP).
 - Algoritmi approssimati a complessità polinomiale (Zaino).
 - Algoritmi euristici (Zaino, TSP).
2. Modellare problemi decisionali che derivano da applicazioni del mondo industriale come problemi di ottimizzazione.

Parte I:

Insiemi indipendenti e coperture

Problema 1: Il ballo (Berge)

- Ad un ballo sono presenti n ragazzi e n ragazze.
- Ciascun ragazzo riconosce k fidanzate tra le ragazze presenti.
- Ciascuna ragazza riconosce k fidanzati tra i ragazzi presenti.

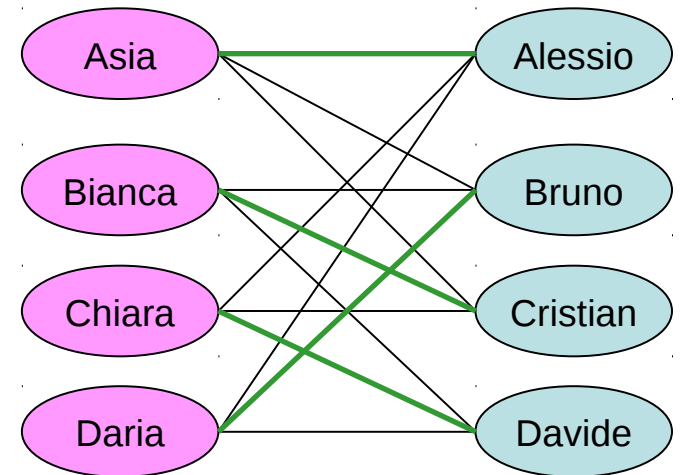
Domanda: è possibile far ballare ciascun ragazzo con una delle sue fidanzate e ciascuna ragazza con uno dei suoi fidanzati?

Problema 1: Il ballo (Berge)

Formulazione: 4 ragazzi/ragazze con 3 fidanzate/fidanzati.

U = sottoinsiemi di archi del grafo

F = famiglia di sottoinsiemi di archi che toccano ogni vertice esattamente una volta.



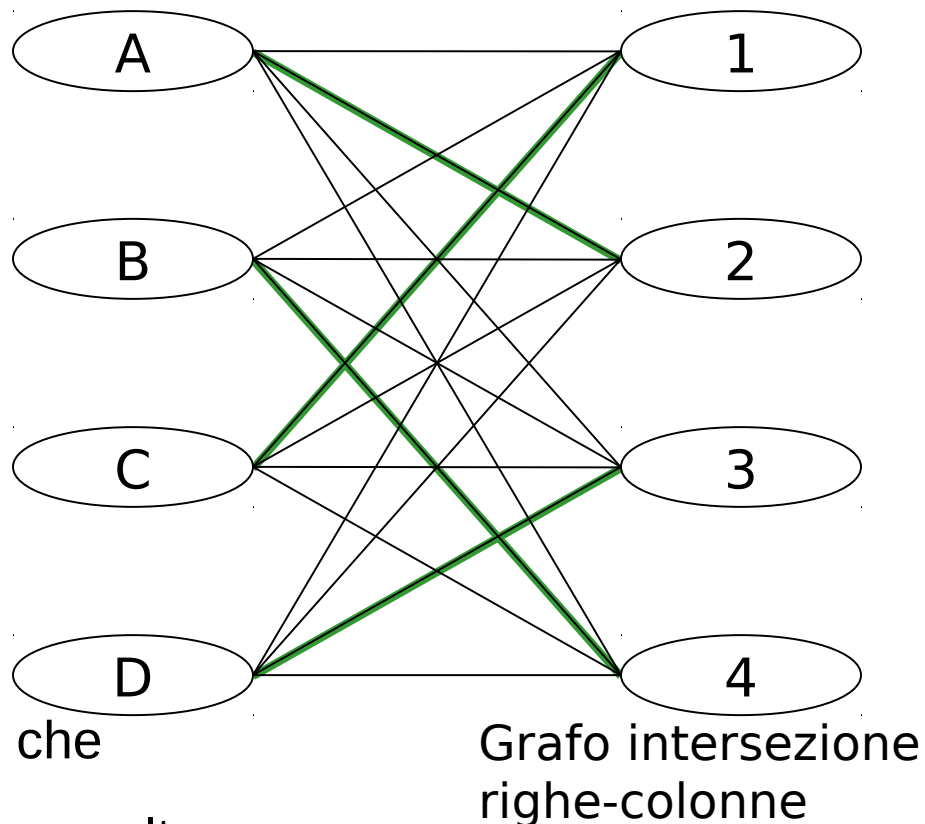
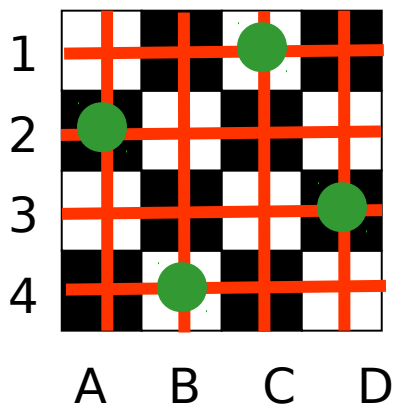
Problema 2: Le torri

- Consideriamo una scacchiera $n \times n$.
- Due torri si danno scacco se giacciono sulla stessa riga (colonna) della scacchiera.

Domanda: Qual è il massimo numero di torri che è possibile disporre sulla scacchiera senza che esse si diano scacco reciproco?

Problema 2: Le torri

Formulazione: Due torri si danno scacco se si trovano sulla stessa riga o colonna.



U = sottoinsiemi di archi del grafo

F = famiglia di sottoinsiemi di archi che toccano ogni vertice non più di una volta.

Problema 3: La battaglia di Inghilterra (Berge)

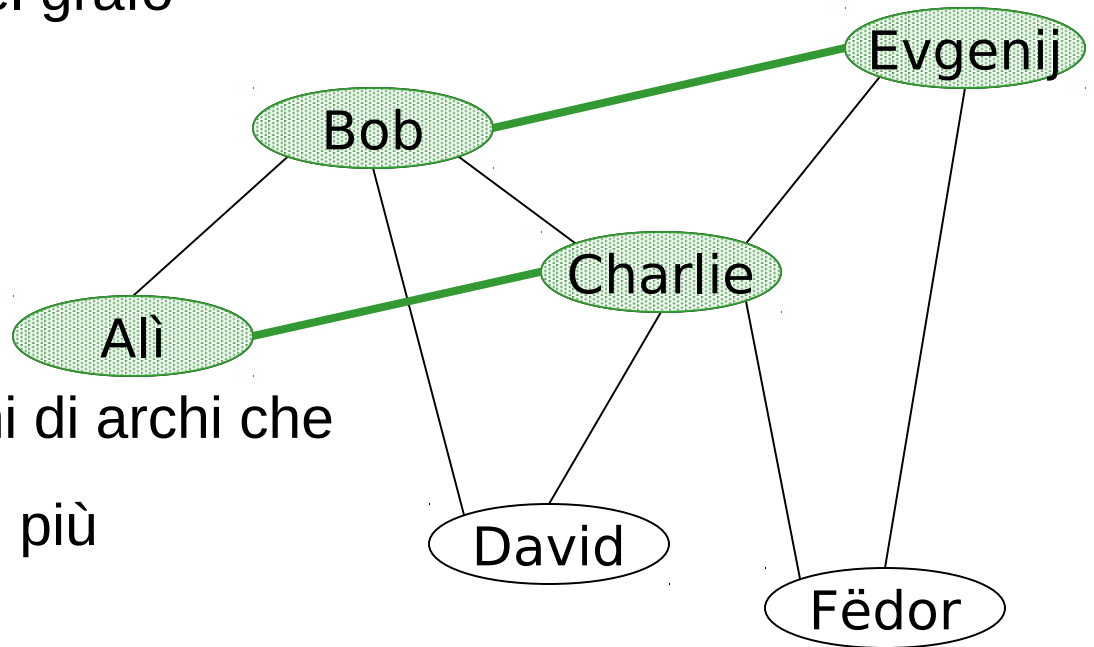
Nel 1941 le squadriglie inglesi erano composte di aerei biposto, ma alcuni piloti non potevano formare una coppia per problemi di lingua o di abitudini.

Domanda: Dati i vincoli di incompatibilità tra coppie di piloti, qual è il massimo numero di aerei che è possibile far volare simultaneamente?

Problema 3: La battaglia di Inghilterra (Berge)

Formulazione: Grafo di compatibilità dei piloti

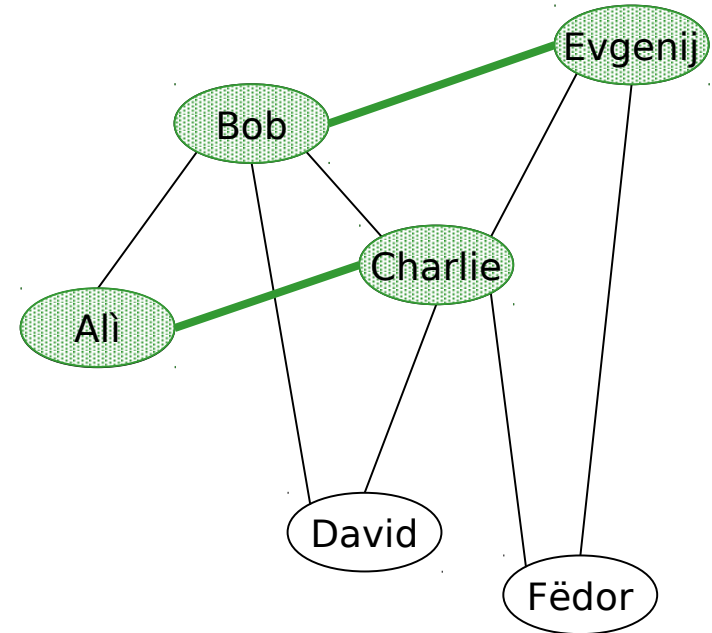
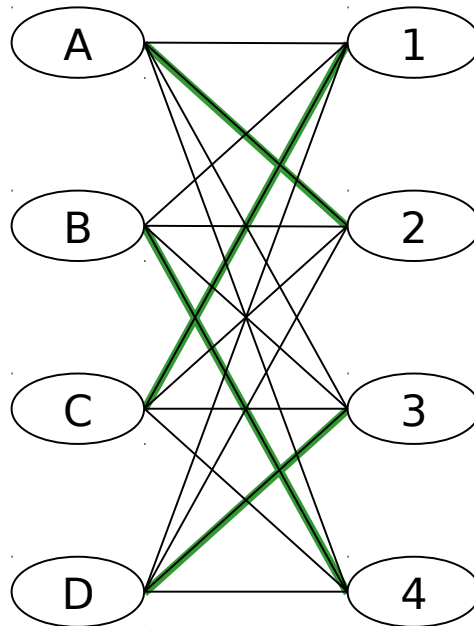
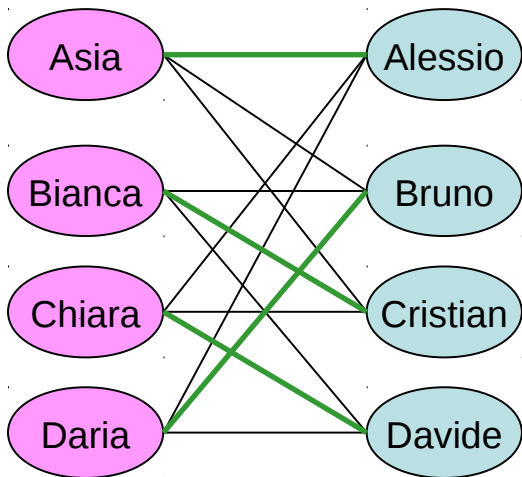
U = sottoinsiemi di archi del grafo



F = famiglia di sottoinsiemi di archi che toccano ogni vertice al più una volta.

Matching

Definizione: Dato un grafo $G = (V, E)$, un *matching* è un sottoinsieme $M \subseteq E$ di archi **a due a due non adiacenti**.



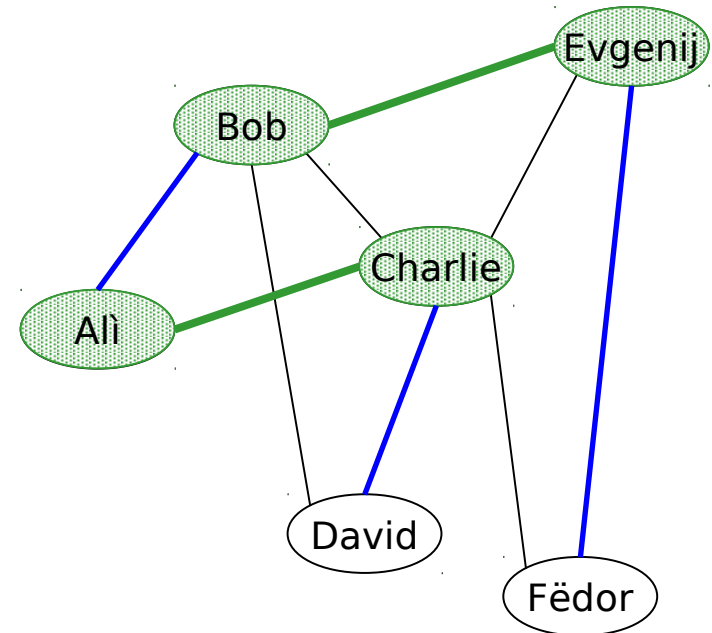
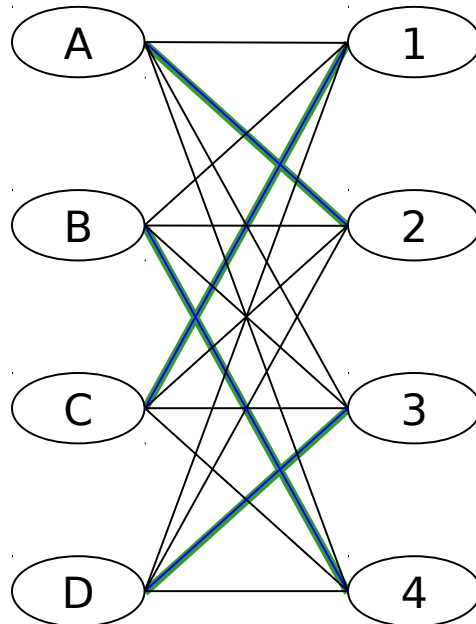
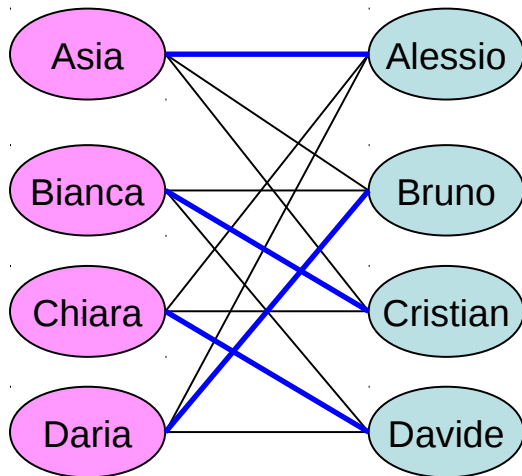
In ciascuno dei problemi precedenti la soluzione corrisponde ad un matching su un grafo.

Tipologie di matching

Definizione: Se $|M^*| \geq |M|$ per ogni matching M di G , allora M^* si dice *massimo*.

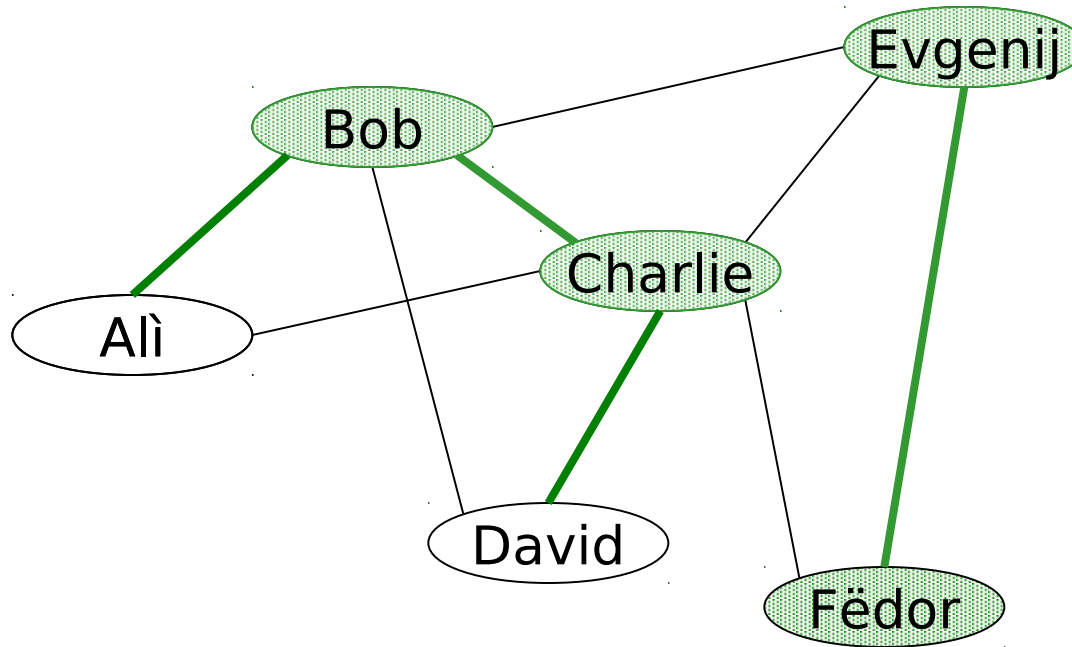
Definizione: Se G è bipartito, allora anche M si dice *bipartito*.

Definizione: Se $|M| = |V|/2$, allora M si dice *perfetto*.



Tipologie di matching

Definizione: Un matching M si dice *massimale* se ogni elemento di $E - M$ è adiacente ad almeno un elemento di M .



Un matching massimale non necessariamente è massimo.

Un matching massimo è sempre massimale.

Insieme indipendente

Definizione: Dato un grafo simmetrico $G = (V, E)$, un qualunque sottoinsieme S di vertici si dice *indipendente* se esso è costituito da elementi *a due a due non adiacenti*.

- S è detto *insieme stabile* (*stable set*)

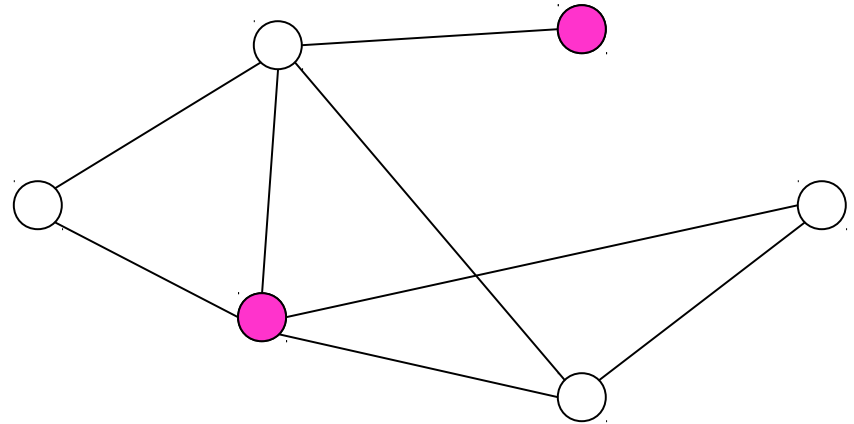
Definizione: Un insieme stabile S^* si dice *massimo* se $|S^*| \geq |S|$, per ogni insieme stabile S di G .

Definizione: Un insieme stabile S si dice *massimale* se ogni elemento di $V - S$ è adiacente ad almeno un elemento di S .

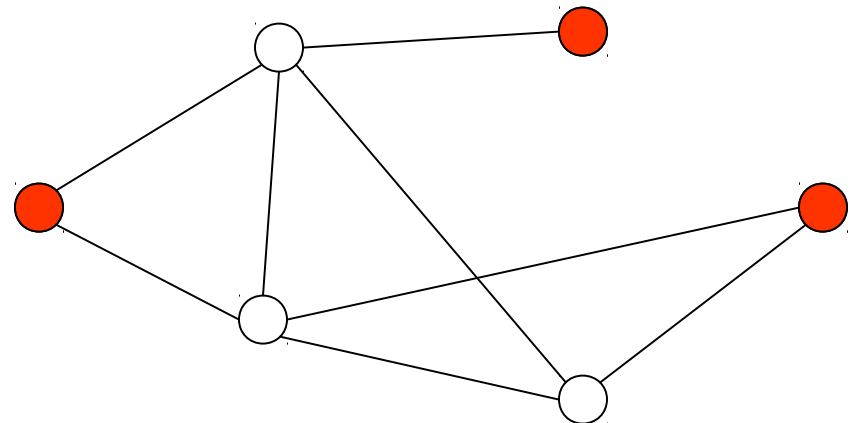
Esempi

Osservazione: L'insieme vuoto è un insieme stabile.

Insieme stabile massimale



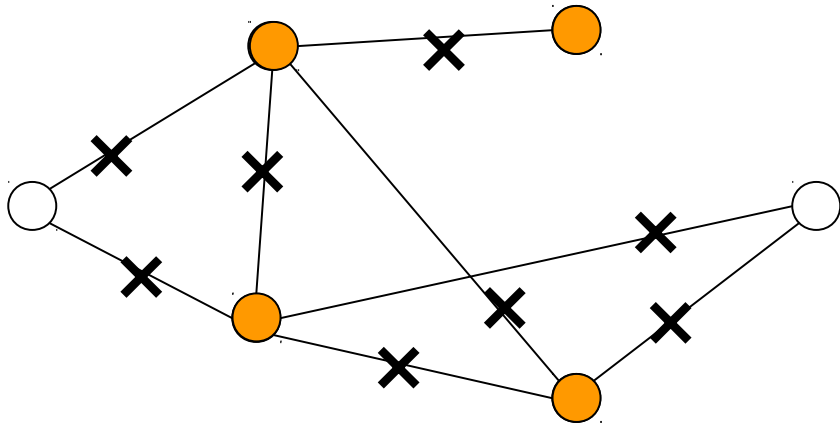
Insieme stabile massimo



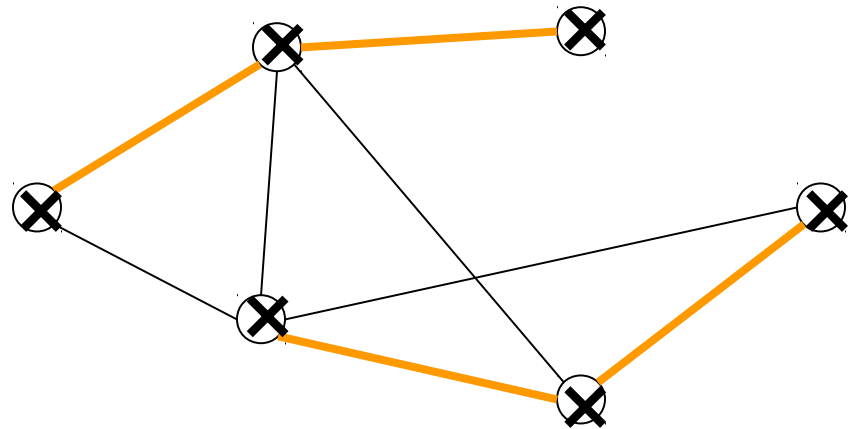
Copertura

Definizione: Dato un grafo simmetrico $G = (V, E)$, un qualunque sottoinsieme T di vertici (F di archi) tale che ogni arco di E (vertice di V) incide su almeno un elemento di T (di F) si dice *copertura*.

- T è detto insieme *trasversale* (*vertex cover*).
- F è detto *edge-cover*.



Trasversale



Edge-cover

Copertura

Definizione: Dato un grafo simmetrico $G = (V, E)$, un qualunque sottoinsieme T di vertici (F di archi) tale che ogni arco di E (vertice di V) incide su almeno un elemento di T (di F) si dice *copertura*.

- T è detto insieme *trasversale* (*vertex cover*).
- F è detto *edge-cover*.

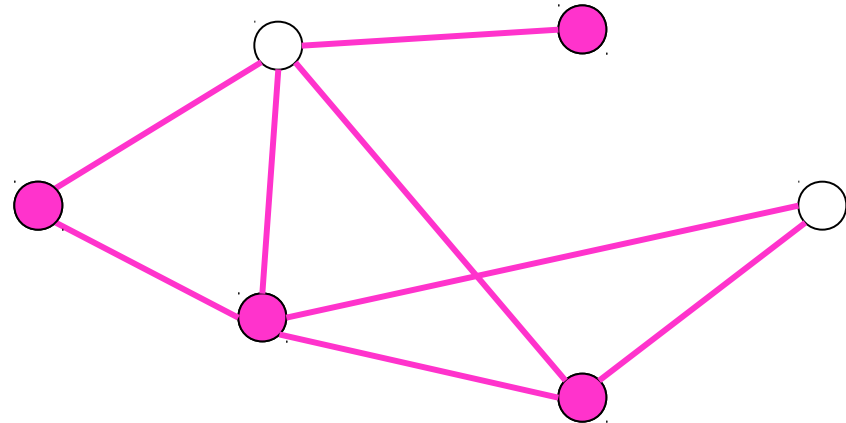
Definizione: Una copertura X si dice *minimale* se $X - \{x\}$ non è una copertura per ogni $x \in X$.

Definizione: Una copertura X^* si dice *minima* se $|X^*| \leq |X|$, per ogni insieme copertura X di G .

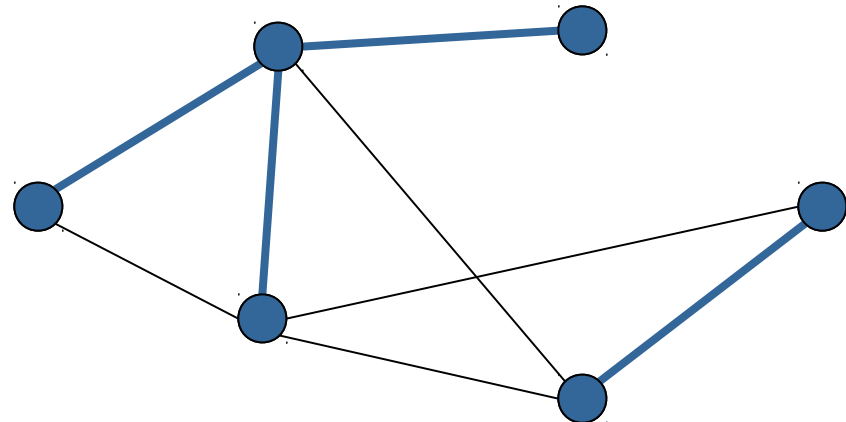
Esempi

Osservazione: L'insieme dei nodi V e l'insieme degli archi E di un grafo sono rispettivamente trasversale e edge-cover.

Trasversale minimale

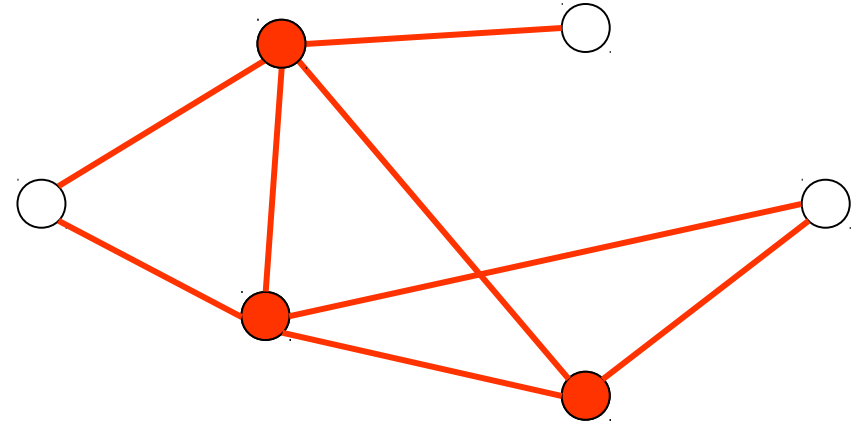


Edge-cover minimale

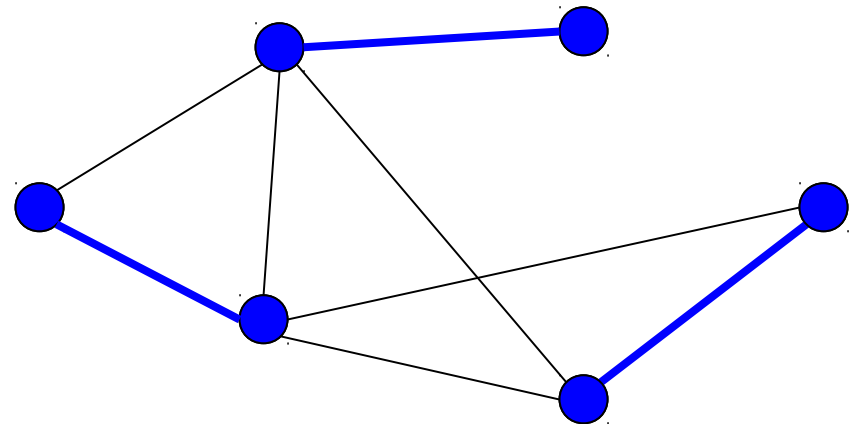


Esempi

Trasversale minimo



Edge-cover minimo



Disuguaglianze duali deboli

Indichiamo con

- $\alpha(G)$ insieme stabile massimo di G .
- $\mu(G)$ matching massimo di G .
- $\rho(G)$ edge cover minimo di G .
- $\tau(G)$ trasversale minimo di G .

Teorema: Per un grafo G valgono le seguenti disuguaglianze

1. $\alpha(G) \leq \rho(G)$

2. $\mu(G) \leq \tau(G)$

Disuguaglianze duali deboli

Dimostrazione: Siano

- X insieme stabile di G , e
- Y edge-cover di G .

Poiché Y copre V , ogni elemento di X incide su almeno un elemento di Y .

D'altra parte, nessun elemento di Y copre contemporaneamente due elementi di X altrimenti i due elementi sarebbero adiacenti e quindi non potrebbero appartenere all'insieme stabile X .

Pertanto, per ogni $x \in X$ esiste un distinto $y \in Y$ che lo copre, e quindi

$$|X| \leq |Y|.$$

Riscrivendo la precedente relazione per gli insiemi massimi X^* e Y^* , si ottiene

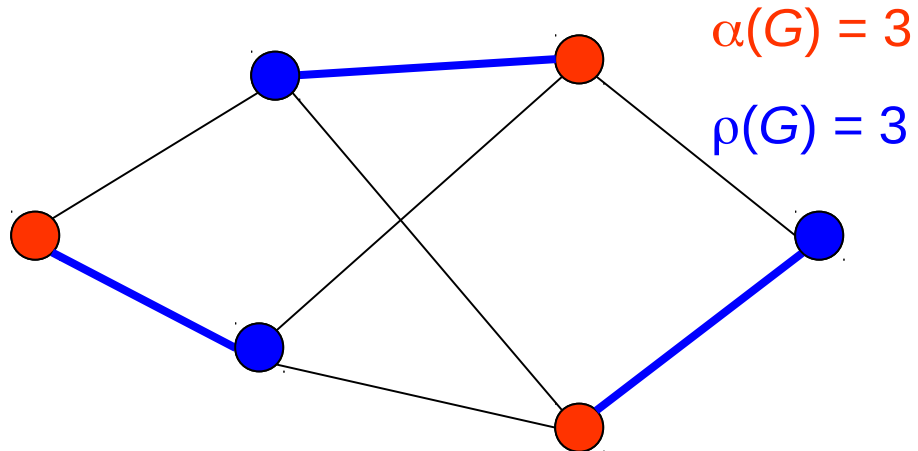
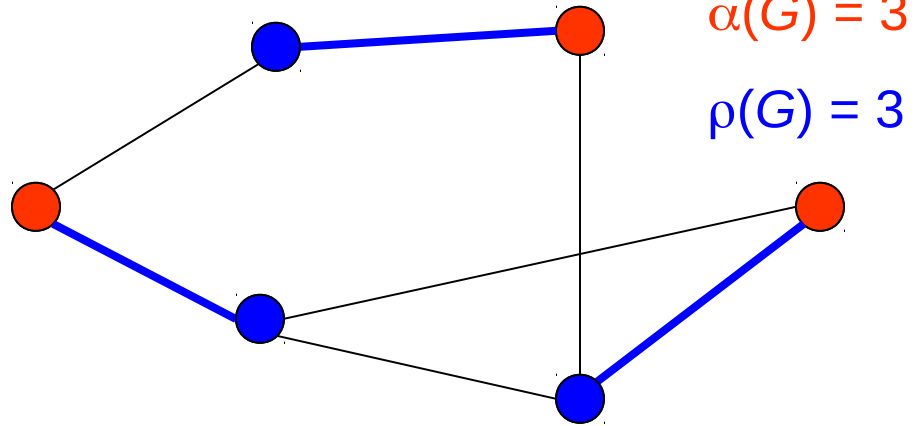
$$\alpha(G) \leq \rho(G)$$

Scambiando il ruolo di V ed E , si ottiene $\mu(G) \leq \tau(G)$

Disuguaglianze duali deboli

Esempio

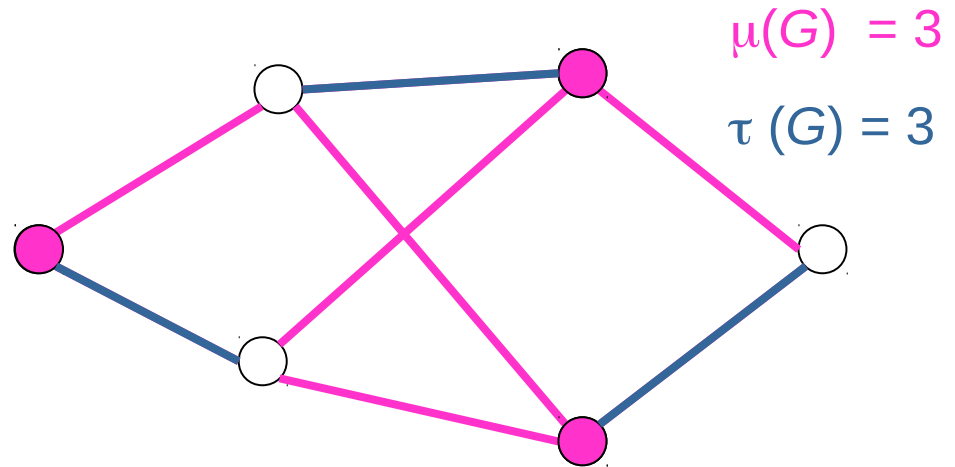
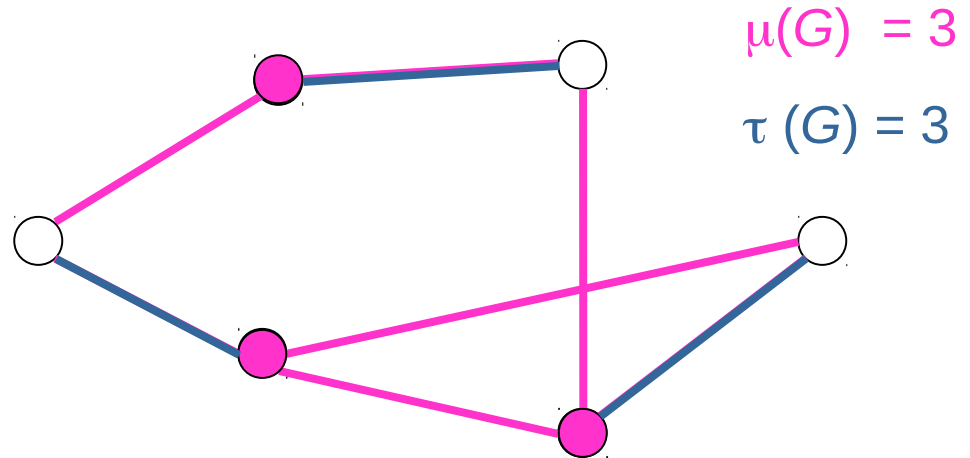
stabile e edge-cover



Disuguaglianze duali deboli

Esempio

trasversale e
matching

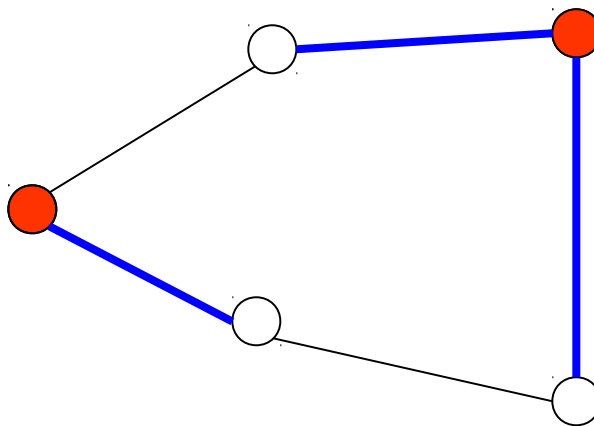


Forse valgono sempre
con il segno “=” ?

Disuguaglianze duali deboli

Forse valgono sempre
con il segno “=” ?

NO!!!



$$\alpha(G) = 2$$

$$\rho(G) = 3$$

Teorema di Gallai

Teorema (Gallai 1959):

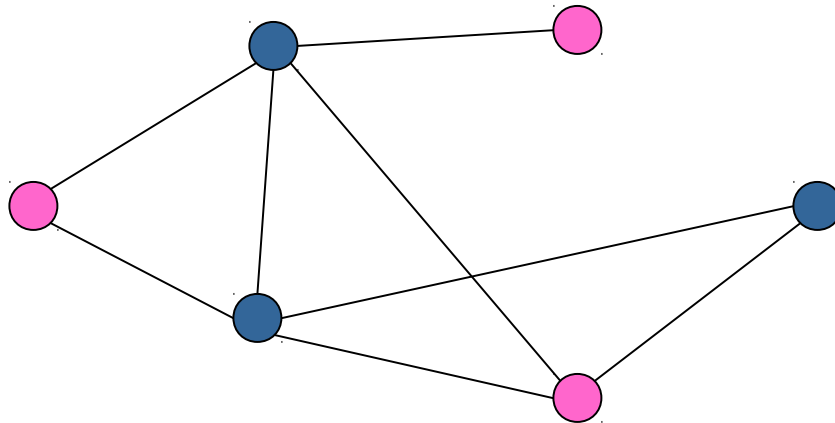
Per ogni grafo G con n nodi si ha:

$$\alpha(G) + \tau(G) = n \quad (1)$$

Se inoltre G non ha nodi isolati

$$\mu(G) + \rho(G) = n \quad (2)$$

Esempio (1)



Teorema di Gallai

Teorema (Gallai 1959):

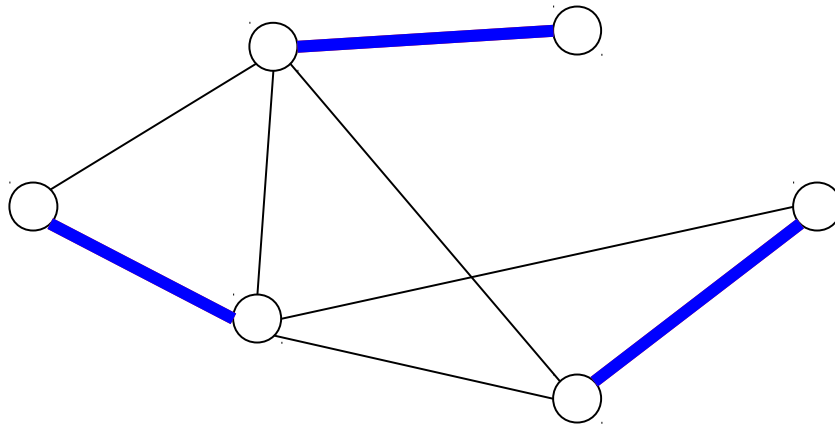
Per ogni grafo G con n nodi si ha:

$$\alpha(G) + \tau(G) = n \quad (1)$$

Se inoltre G non ha nodi isolati

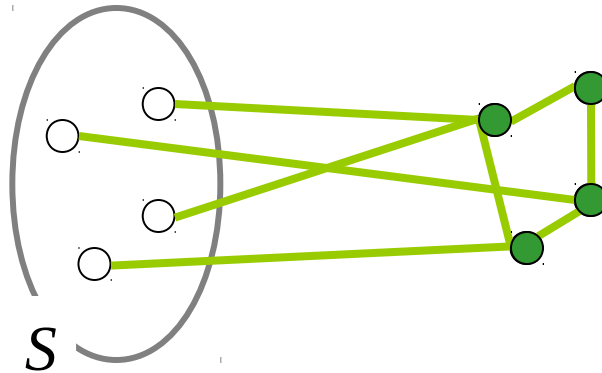
$$\mu(G) + \rho(G) = n \quad (2)$$

Esempio (2)



Dimostrazione

Dimostrazione (1): Sia S un insieme stabile di G . Allora $V-S$ è un insieme trasversale.



In particolare, $|V-S| \geq \tau(G)$.

Se consideriamo l'insieme stabile massimo S^* , otteniamo

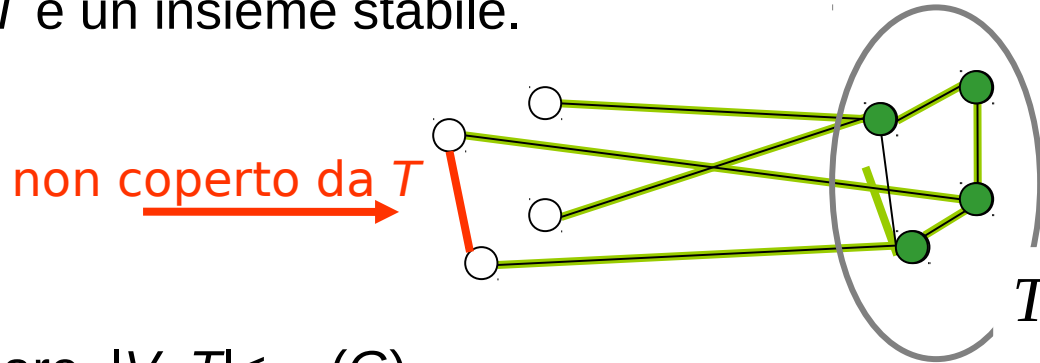
$$\tau(G) \leq |V-S^*| = n - \alpha(G)$$

da cui ricaviamo

$$\alpha(G) + \tau(G) \leq n.$$

Dimostrazione

Dimostrazione (1): Viceversa, sia T un insieme trasversale di G . Allora $V-T$ è un insieme stabile.



In particolare, $|V-T| \leq \alpha(G)$.

Se consideriamo l'insieme trasversale minimo T^* , otteniamo

$$\alpha(G) \geq |V-T^*| = n - \tau(G)$$

da cui ricaviamo

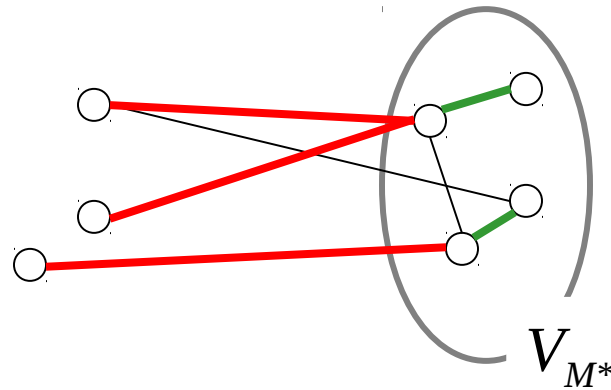
$$\alpha(G) + \tau(G) \geq n.$$

Considerando la condizione ottenuta precedentemente, possiamo concludere che

$$\alpha(G) + \tau(G) = n.$$

Dimostrazione

Dimostrazione (2): Sia G un grafo privo di nodi isolati e sia M^* il **matching massimo** di G . Indichiamo con V_{M^*} i nodi che sono estremi degli archi in M^* .



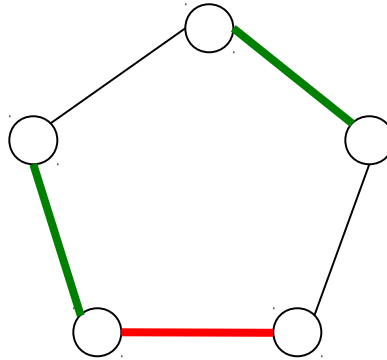
Sia H un **insieme minimale** di archi tale che ogni nodo in $V - V_{M^*}$ è estremo di qualche arco in H .

Segue che $|H| = |V - V_{M^*}| = n - 2|M^*|$

Osserviamo che l'insieme $C = H \cup M^*$ è un edge-cover di G .

Dimostrazione

Dimostrazione (2):



Sicuramente, $|C| \geq \rho(G)$.

Quindi

$$\rho(G) \leq |C| = |M^*| + |H| = |M^*| + n - 2|M^*| = n - |M^*| = n - \mu(G)$$

da cui ricaviamo $\rho(G) + \mu(G) \leq n$.

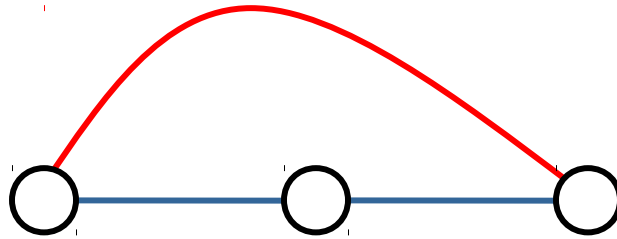
Dimostrazione

Dimostrazione (2): Sia C il minimo edge-cover su G ($|C| = \rho(G)$).

Sia $H = (V, C)$ il sottografo indotto da C .

Valgono le seguenti proprietà:

1) H è un grafo aciclico.

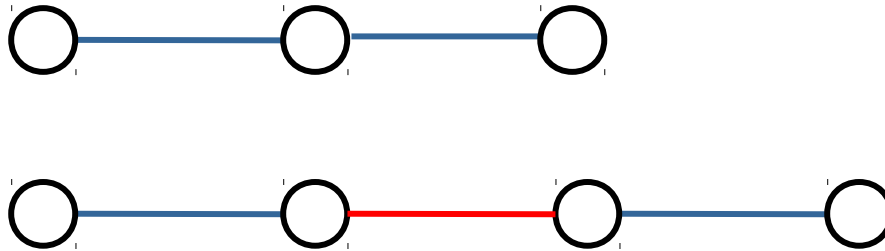


Infatti, se H contenesse cicli allora C non sarebbe un edge-cover minimo.

Dimostrazione

Dimostrazione (2):

2) Ogni cammino in H è composto di al più due archi.



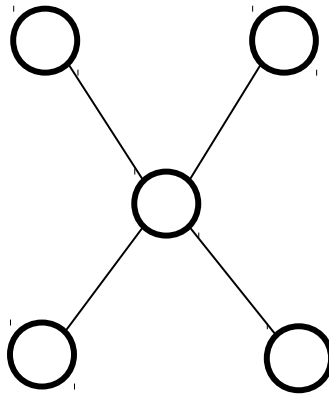
Infatti, se H contenesse un cammino con 3 archi sarebbe sempre possibile rimuovere un arco e ottenere ancora un edge-cover. L'esistenza di un tale cammino contraddirebbe il fatto che C è minimo.

Dimostrazione

Dimostrazione (2):

Dalle proprietà precedenti concludiamo che il grafo $H = (V, C)$

- ha $|V| = n$ vertici;
- ha $|C| = \rho(G)$ archi;
- può essere decomposto in N componenti connesse aventi la forma di “stella”



Dimostrazione

Dimostrazione (2):

Consideriamo l' i -esima componente connessa di H . Indichiamo con:

- s_i = numero di nodi della componente connessa, e
- $s_i - 1$ = numero di archi della componente connessa.

Pertanto

$$n = \sum_{i=1}^N s_i \quad \text{e} \quad \rho(G) = \sum_{i=1}^N (s_i - 1) = n - N \Rightarrow N = n - \rho(G)$$

Sia M un matching con un arco per ogni componente di H . Si ottiene

$$\mu(G) \geq |M| = n - \rho(G) \Rightarrow \rho(G) + \mu(G) \geq n$$

Considerando la condizione ottenuta precedentemente, possiamo concludere che

$$\rho(G) + \mu(G) = n.$$

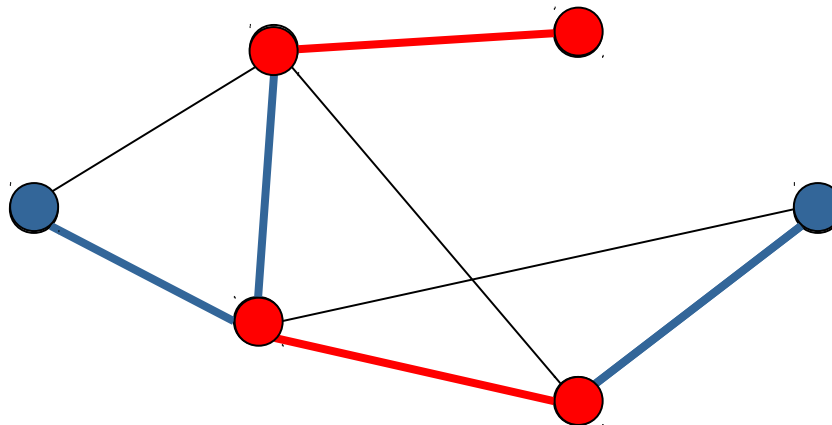
Cammino alternante

Sia M un matching di $G = (V, E)$.

Definizione: Un arco $(i, j) \in E$ si dice *accoppiato* (*libero*) se $(i, j) \in M$ ($(i, j) \notin M$).

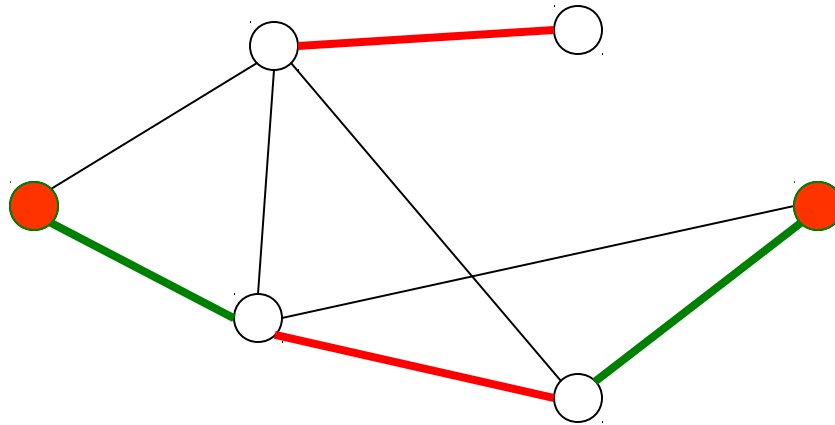
Definizione: Un vertice $i \in V$ si dice *accoppiato* (*esposto*) se su di esso *incide* (*non incide*) un arco di M .

Definizione: Un cammino P sul grafo G si dice *alternante* rispetto a M se esso è costituito alternativamente da archi accoppiati e liberi.



Cammino aumentante

Definizione: Un cammino P alternante rispetto ad M che abbia entrambi gli estremi esposti si dice *aumentante*.

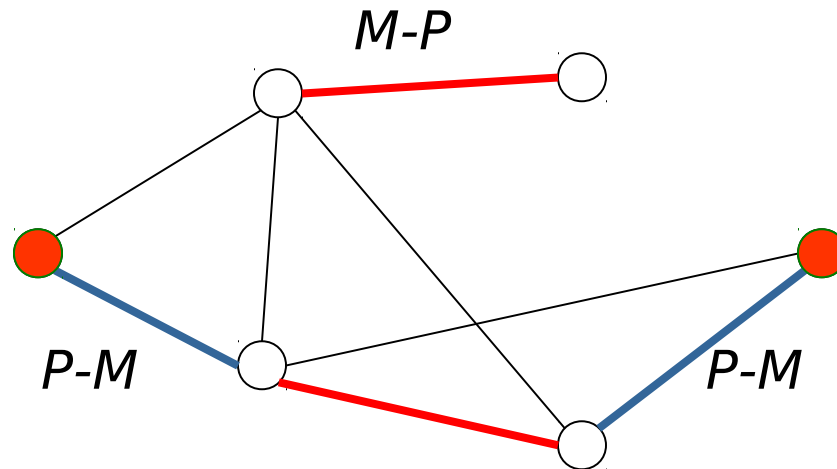


Aumentare un matching

Teorema: Sia M un matching di G e sia P un cammino aumentante rispetto a M . La differenza simmetrica

$$M' = (M - P) \cup (P - M) = M \oplus P$$

è un matching di cardinalità $|M| + 1$.



Dimostrazione

Dimostrazione: Sia M un matching di G e sia P un cammino aumentante rispetto a M . L'insieme

$$M' = (M - P) \cup (P - M)$$

gode delle seguenti proprietà

1) M' è un matching. Infatti, se così non fosse allora esisterebbe almeno un nodo in cui incidono due archi di M' .

Osserviamo che:

1) Per i nodi che non sono toccati da P non è cambiato nulla. Infatti, su essi incideva ed incide un solo arco di M che ora appartiene anche ad M' .

2) Sui nodi intermedi di P prima incideva un arco di M e adesso incide soltanto un arco di $P - M$ (e quindi di M').

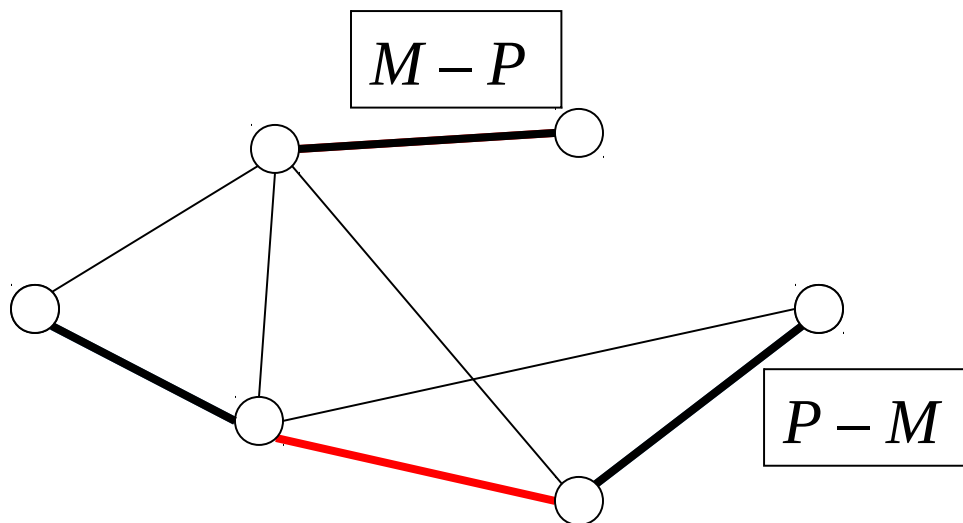
Dimostrazione

3) I nodi estremi di P prima erano esposti e adesso sono accoppiati e su di essi incide soltanto un arco di $P - M$ (e quindi di M').

Pertanto possiamo concludere che M' è effettivamente un matching.

Dimostrazione

Dimostrazione: 2) M' ha un elemento in più di M .



Sia $|M| = m_1 + m_2$ con $m_1 = |M - P|$ e $m_2 =$ numero di archi del matching appartenenti al cammino.

Poiché P è aumentante, $|P| = m_2 + (m_2 + 1)$ dove $(m_2 + 1) = |P - M|$.

Pertanto: $|M'| = |M - P| + |P - M| = m_1 + m_2 + 1 = |M| + 1$.

Teorema di Berge

Teorema (Berge,1957): Un matching M di G è massimo se e solo se G non ammette cammini aumentanti rispetto a M .

Dimostrazione (\Rightarrow): Segue direttamente dal teorema precedente.

Dimostrazione (\Leftarrow): Facciamo vedere che, se non esistono cammini aumentanti rispetto a un certo matching M , allora quel matching M è massimo.

Supponiamo che G ammetta un matching M' con un elemento in più di M .

Vogliamo dimostrare che allora esiste un cammino aumentante per M .

Teorema di Berge

Dimostrazione (\Rightarrow): Consideriamo l'insieme di archi

$$F = M' \oplus M$$

e sia G' il sottografo di G avente gli stessi nodi di G ma contenente solo l'insieme di archi in F .

Analizziamo il grado di ciascun nodo di G' , considerando tutti i casi possibili.

1. Un nodo su cui incide lo stesso arco appartenente sia ad M che ad M' è un nodo isolato su G' e quindi ha grado 0.
2. Un nodo su cui incide sia un arco di M sia un arco di M' è un nodo che ha grado 2 su G' .
3. Un nodo su cui incide un arco di M e nessun arco di M' o viceversa è un nodo che ha grado 1 su G' .

Teorema di Berge

Dimostrazione (\Rightarrow): Consideriamo l'insieme di archi

$$F = M' \oplus M$$

e sia G' il sottografo di G individuato dall'insieme di archi F e da tutti i loro estremi.

Analizziamo il grado di ciascun nodo di G' , considerando tutti i casi possibili.

4. Un nodo esposto sia rispetto ad M che rispetto ad M' è un nodo isolato su G' e quindi ha grado 0.

Pertanto in G' nessun nodo ha grado superiore a 2 e possiamo concludere che le componenti connesse di G' sono o nodi isolati o percorsi o cicli.

Teorema di Berge

Dimostrazione (\Rightarrow): Nessun ciclo può essere dispari altrimenti ci sarebbero due archi dello stesso matching incidenti sullo stesso nodo e questo è impossibile.

Non possono essere tutti cicli pari altrimenti $|M| = |M'|$. Deve esistere una componente connessa che è un percorso.

Non tutti i percorsi possono essere pari altrimenti $|M| = |M'|$.

Quindi, senza perdita di generalità, possiamo assumere che esista un percorso dispari che inizia e termina con un arco di M' .

Questo percorso è aumentante per M .

Un possibile algoritmo

```
M =  $\emptyset$ ; // Inizializzazione
trovato = TRUE;
while (trovato) {
    search (G, M, &trovato);
    if (trovato)
        aumenta (G, &M);
}
```

Come è fatta **search (*G*, *M*, &*trovato*)**?

Teorema del cammino aumentante

Teorema: Sia v un vertice esposto in un matching M . Se non esiste un cammino aumentante per M che parte da v , allora esiste un matching massimo avente v esposto.

Dimostrazione: Sia M^* un matching massimo in cui v è accoppiato.

Consideriamo $M \oplus M^*$.

Dimostrazione

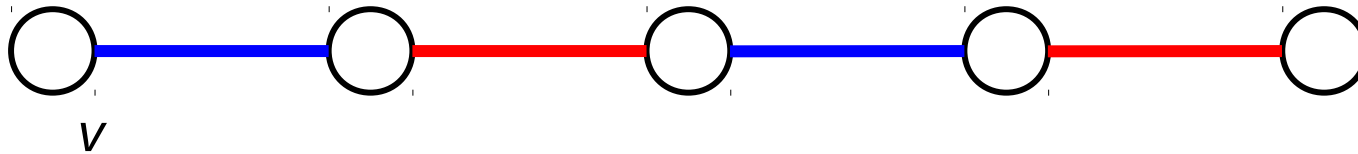
Dimostrazione:

$M \oplus M^*$ non può contenere un cammino



perché aumentante per M .

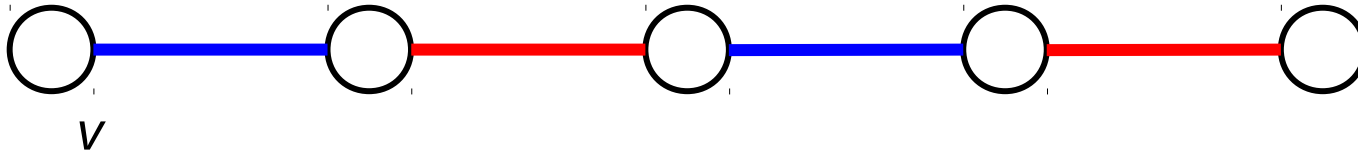
Però deve contenere un cammino composto dallo stesso numero di archi di M e di M^*



Infatti, un cammino con un solo arco di M (di M^*) sarebbe aumentante per M^* (per M)

Dimostrazione

Indichiamo con P il cammino



e consideriamo un nuovo matching $M' = M^* \oplus P$.

Osserviamo che

- $|M'| = |M^*|$
- il nodo v è esposto rispetto ad M' .

Pertanto abbiamo individuato un nuovo matching massimo con v esposto.

Un possibile algoritmo (II)

```
M =  $\emptyset$ ; trovato = FALSE; //Inizializzazione
for (v ∈ V) {
    if (v è esposto) {
        search (v, M, &trovato, &q);
        if (trovato)
            aumenta (q, v, &M);
        else
            //cancella v e tutti gli archi
            incidenti in v
            cancella (v, &G);
    }
}
```

Ricerca di cammini aumentanti

Scopo della funzione *search*:

trovare un cammino aumentante rispetto a M , oppure dire che non esiste.

Parametri

v = nodo esposto;

M = matching;

trovato = variabile booleana;

q = vertice estremo del cammino aumentante.

Introduciamo un'etichetta per i vertici di V

label(w) = PARI
= DISPARI
= NULL

Ricerca di cammini aumentanti

```
search (v, M, *q, *trovato) {  
    for (i ∈ V)  
        label (i) = NULL;  
    LIST = {v};  
    label {v} = PARI;  
    while (LIST != ∅) {  
        pop (&i, LIST);  
        if (label (i) == PARI)  
            esplora_pari (i, M, q, trovato, &LIST);  
        else  
            esplora_dispari (i, M, &LIST);  
        if (trovato)  
            return;  
    }  
}
```

Ricerca di cammini aumentanti (II)

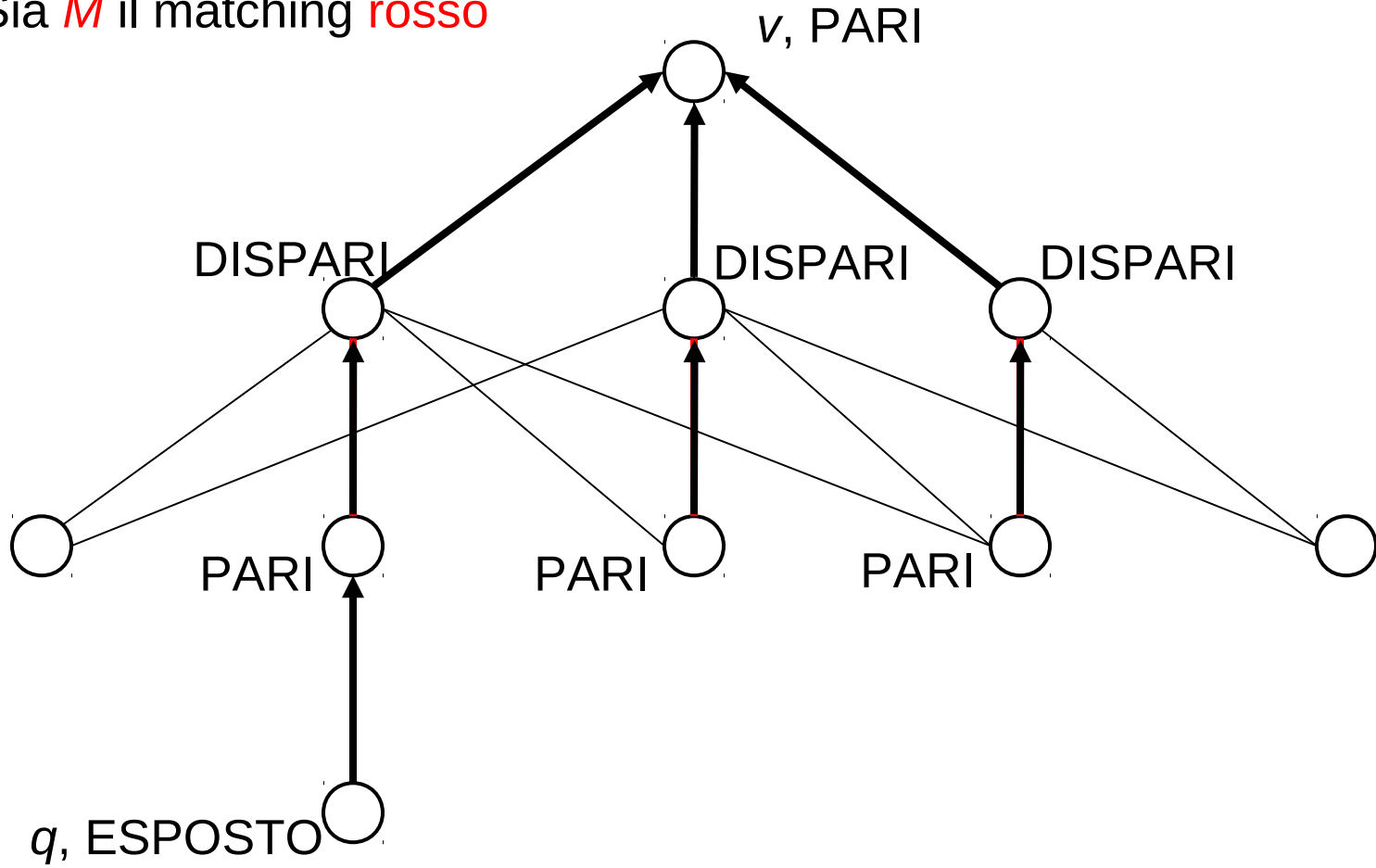
```
esplora_pari (i, M, *q, *trovato, *LIST) {  
    for (j  $\in$   $\delta(i)$ ) {  
        if (j  $\notin$  M) {  
            *q = j;  
            *trovato = TRUE;  
            pred (q) = i;  
            return;  
        }  
        if (j  $\in$  M && label (j) == NULL) {  
            pred (j) = i;  
            label (j) = DISPARI;  
            push (j, LIST);  
        }  
    }  
}
```

Ricerca di cammini aumentanti (III)

```
esplora_dispari (i, M, *LIST)  
{  
  j = vertice accoppiato ad i in M;  
  if (label (j) == NULL){  
    pred (j) = i;  
    label (j) = PARI;  
    push (j, LIST);  
  }  
}
```

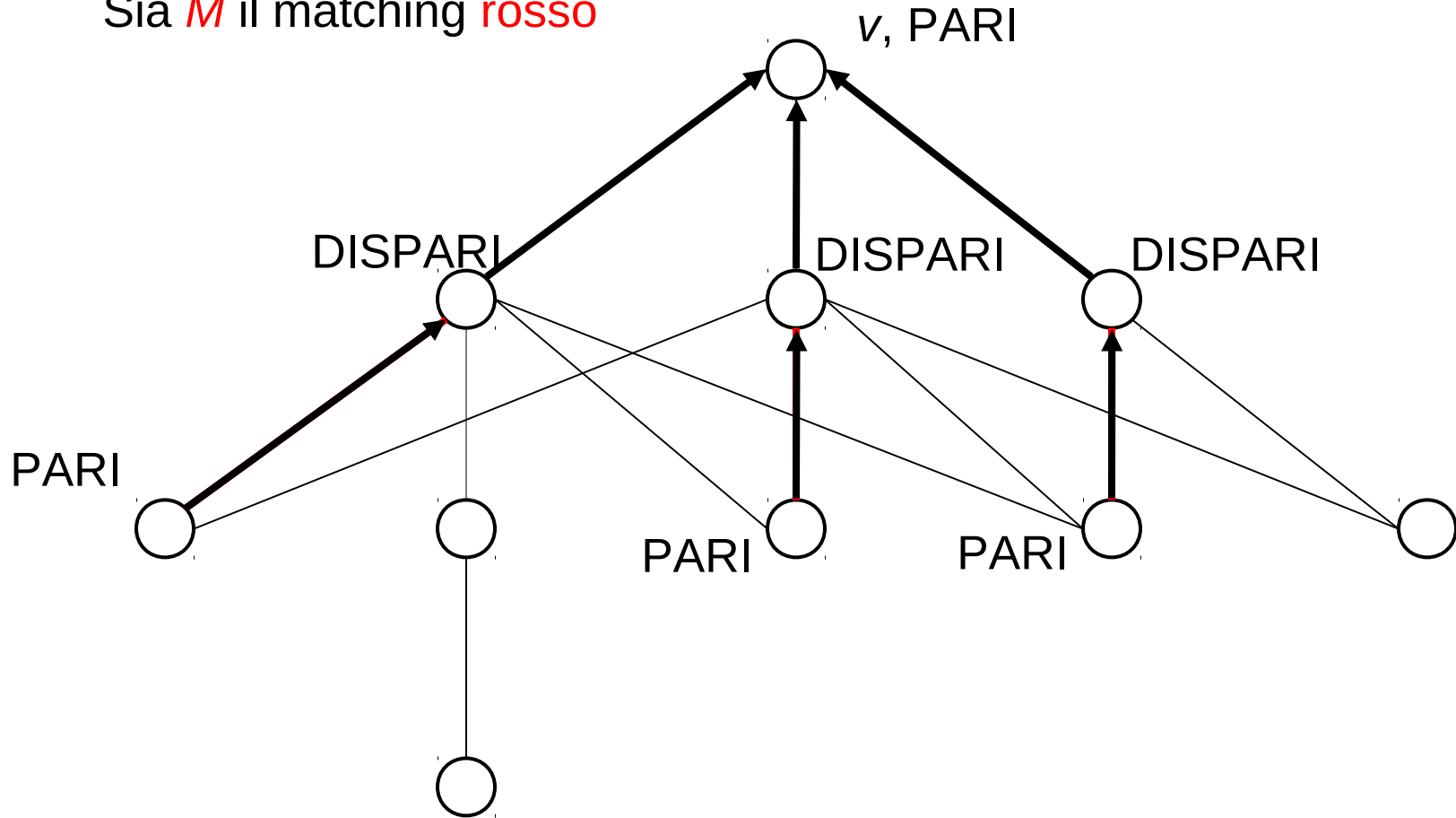
Esempio

Sia M il matching rosso

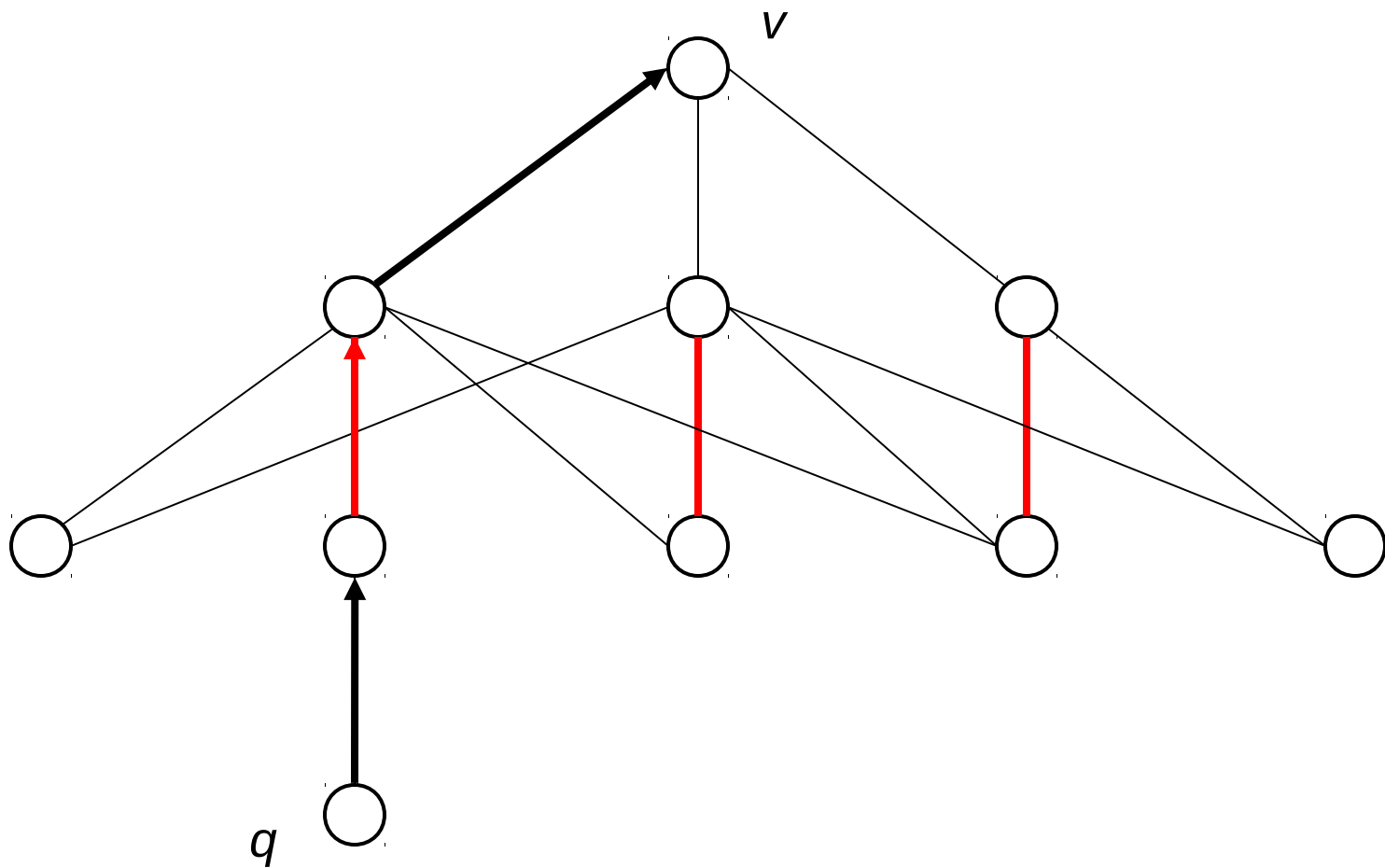


Esempio

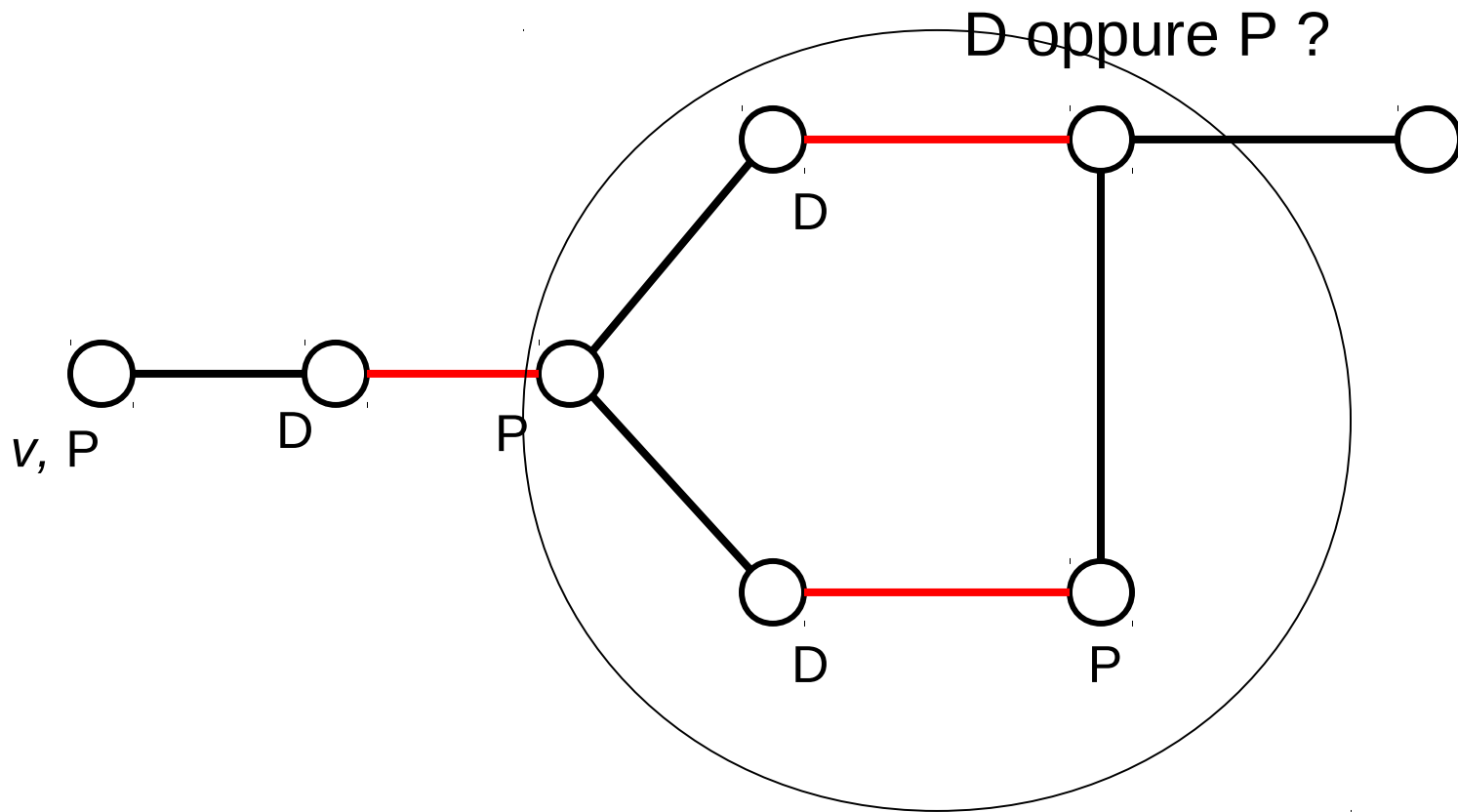
Sia M il matching rosso



Esempio



Un problema



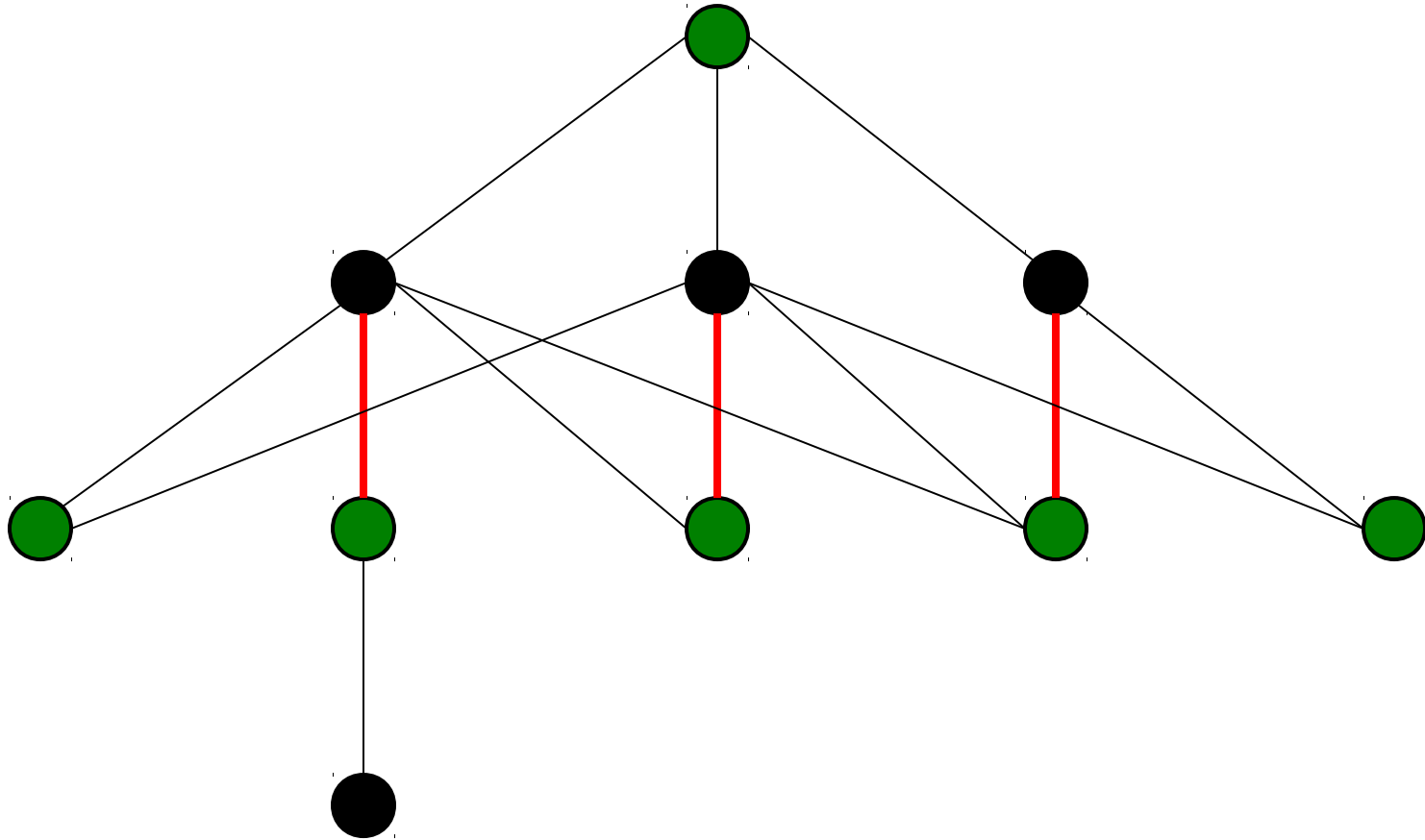
Correttezza

Teorema: Se i vertici di G sono etichettati in modo “unico” dalla procedura `search` rispetto ad un matching M , allora `search` termina con un cammino aumentante, se esso esiste.

Domanda: Esistono grafi che ammettono sempre la proprietà di unicità delle etichette?

Grafi bipartiti

Grafo Bipartito $G=(X, Y, E)$



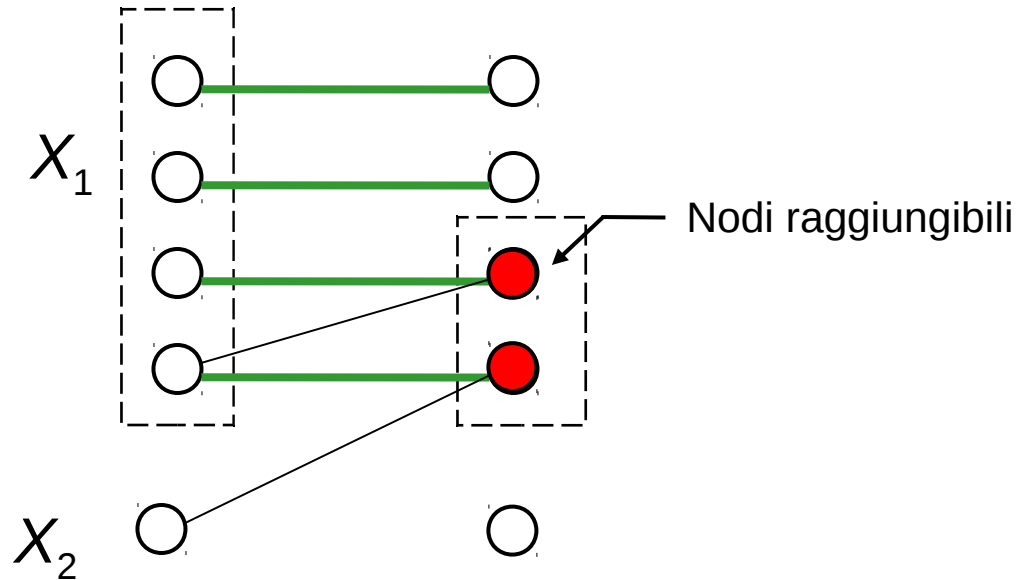
Teorema di König

Teorema (König 1931): Se $G = (X, Y, E)$ è un grafo bipartito allora $\mu(G) = \tau(G)$.

Dimostrazione: Sia M^* un matching massimo, e siano

- X_1 : insieme dei nodi x di X accoppiati rispetto ad M^* .
- X_2 : insieme dei nodi x di X esposti rispetto ad M^* .

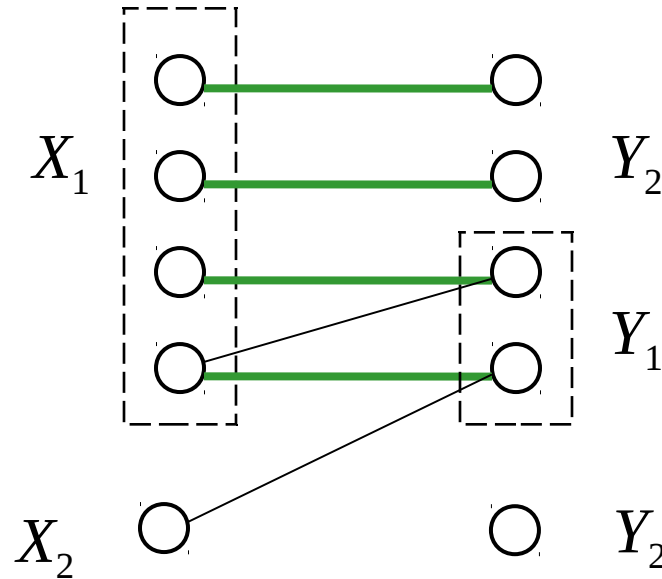
Dimostrazione



Definizione: Un nodo $y \in Y$ è **raggiungibile** se esiste P alternante rispetto ad M^* da x in X_2 tale che l'ultimo arco **non appartiene** ad M^* .

Y_1 : insiemi dei nodi y di Y raggiungibili da x in X_2 .

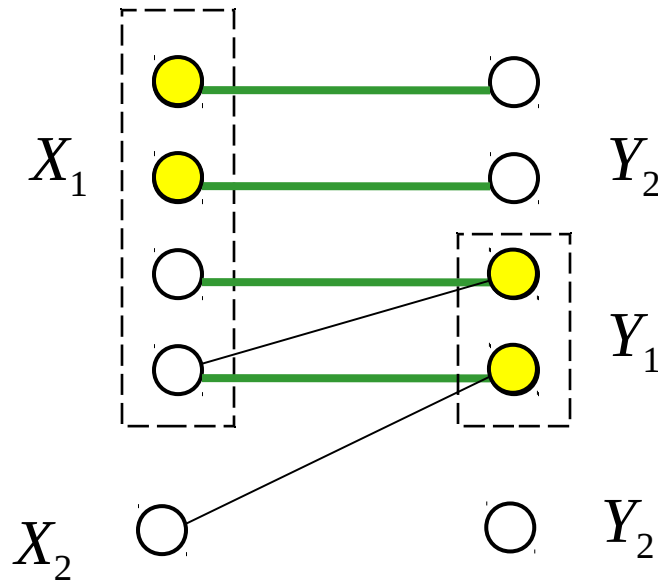
Dimostrazione



Osservazione: Per definizione i nodi in Y_1 sono accoppiati, altrimenti M^* non sarebbe massimo.

Infine: $Y_2: Y - Y_1$

Dimostrazione



Consideriamo il seguente insieme di nodi

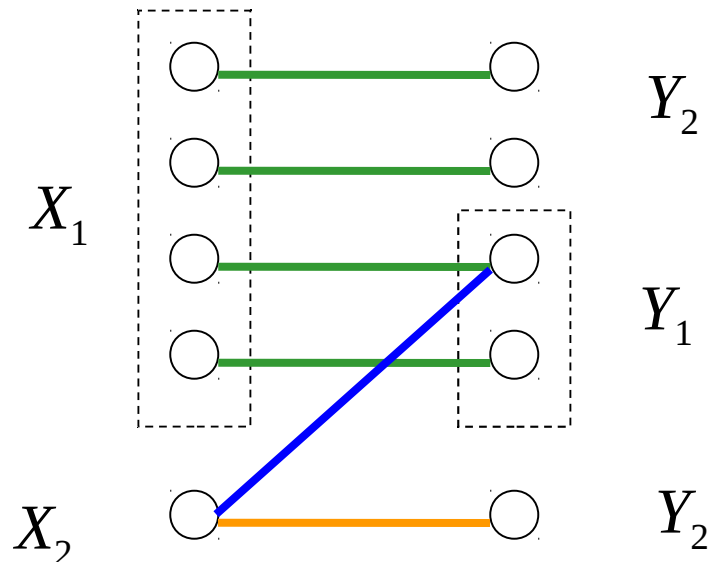
$$Z = \{z_1, z_2, \dots, z_{u(G)}\} \quad \text{con} \quad \begin{array}{ll} z_i = y_i & \text{se } y_i \text{ è raggiungibile} \\ z_i = x_i & \text{altrimenti} \end{array}$$

e dimostriamo che è un trasversale.

Dimostrazione

Dimostriamo che non esistono archi da nodi in X_2 verso nodi in Y non coperti da Z . Infatti,

- 1) Non può esistere un arco non coperto da Z tra un nodo in X_2 e un nodo in Y_2 , altrimenti il matching non sarebbe massimo.
- 2) Non può esistere un arco non coperto da Z tra un nodo in X_2 e un nodo in Y_1 perché i nodi in Y_1 sono raggiungibili e quindi l'arco necessariamente deve essere coperto.

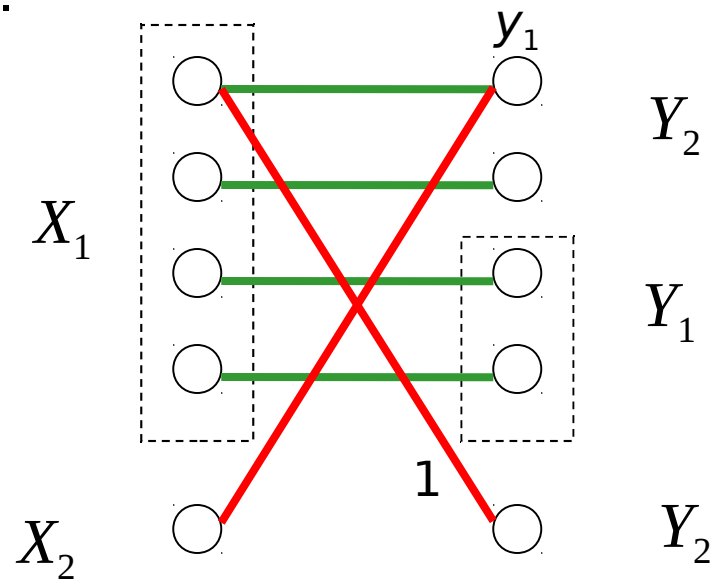


Dimostrazione

Dimostriamo che non esistono archi da nodi in X_1 verso nodi in Y non coperti da Z .

Consideriamo l'arco 1, da X_1 a Y_2 . Se non fosse coperto allora il nodo y_1 , estremo dell'arco del matching, sarebbe raggiungibile.

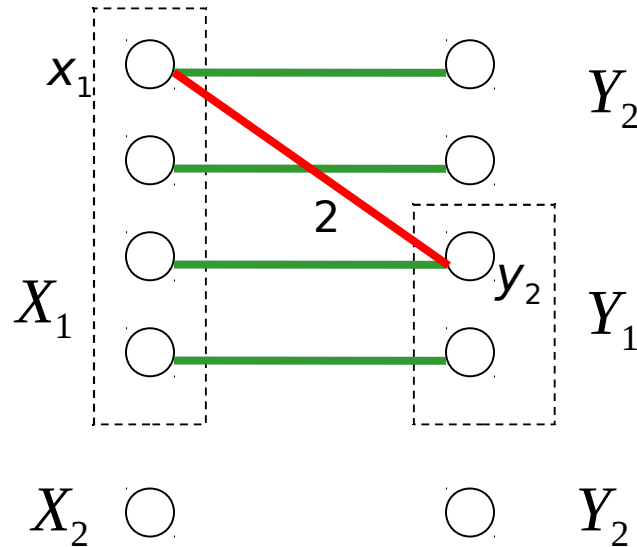
Ma allora esisterebbe un cammino aumentante e il matching non sarebbe massimo.



Dimostrazione

Consideriamo un arco da x_1 a y_1 , ad esempio l'arco 2.

Se non fosse coperto allora il nodo y_2 non sarebbe raggiungibile, ovvero non apparterebbe ad Y_1 (contraddizione).



Pertanto Z è un trasversale di cardinalità pari a $\mu(G)$.

Matroidi

L'algoritmo Greedy

Ricordiamo la struttura di un problema di Ottimizzazione Combinatoria:

- $N = \{1, 2, \dots, n\}$ insieme finito.
- c vettore di pesi (costi/profitti) con coordinate c_j definite per ogni $j \in N$
- Insieme universo: $U = \{\text{tutti i possibili } 2^{|N|} \text{ sottoinsiemi di } N\}$
- Famiglia ammissibile: $F = \{\text{sottoinsiemi } F \text{ di } U \text{ che soddisfano una certa proprietà } P\}$.

Problema di ottimizzazione combinatoria (in forma di minimo)

$$\min_{S \subseteq N} \left\{ \sum_{j \in S} c_j : S \in F \right\}$$

L'algoritmo Greedy

Abbiamo visto come l'algoritmo di enumerazione sia inapplicabile nella pratica per risolvere un problema di Ottimizzazione Combinatoria.

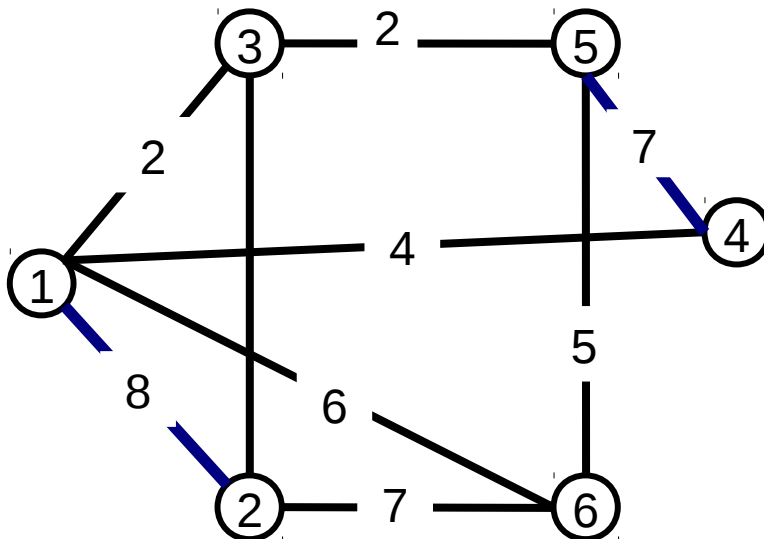
Un algoritmo meno complesso è il così detto *algoritmo greedy* che può essere descritto come segue (per un problema in forma di minimo):

```
S =  $\emptyset$ ; // Inizializzazione
while (U  $\neq \emptyset$ ) {
    Scegli u  $\in$  U tale che c(u)  $\leq$  c(x) per ogni x  $\in$  U
    if (S  $\cup$  {u}  $\in$  F)
        S = S  $\cup$  {u};
        U = U / {u};
}
```

L'algoritmo Greedy

Domanda: L'algoritmo greedy fornisce SEMPRE una soluzione AMMISSIBILE per un problema di Ottimizzazione Combinatoria?

Consideriamo il problema di determinare il matching perfetto di peso massimo su un grafo G .



Soluzione greedy: $M^* = \{(1,2), (4,5)\}$

Non è una soluzione ammissibile perché il matching non è perfetto!

Problemi subclusivi

Definizione: Un problema di Ottimizzazione Combinatoria si dice subclusivo se la famiglia F soddisfa le seguenti condizioni:

1. $\emptyset \in F$
2. $\forall F \in F, Y \subseteq F \Rightarrow Y \in F.$

L'algoritmo greedy restituisce sempre una soluzione ammissibile se il problema è subclusivo.

Il problema del matching perfetto NON E' SUBCLUSIVO.

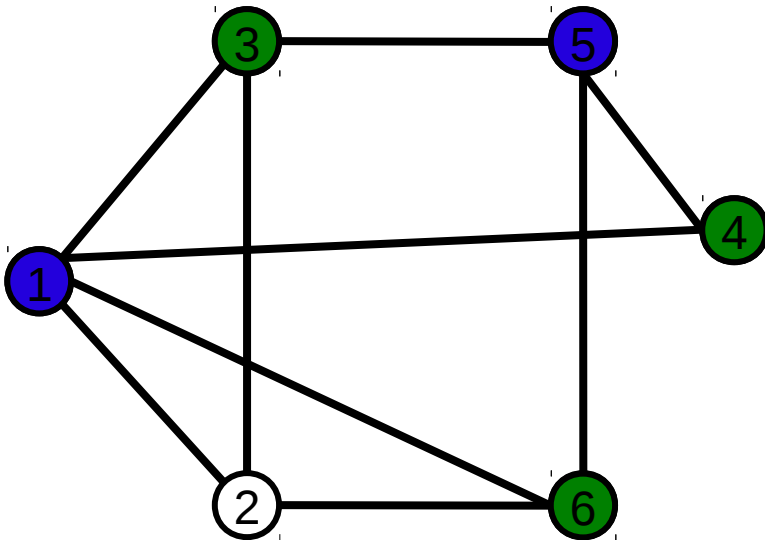
Il problema del matching (non perfetto) E' SUBCLUSIVO.

Il problema dell'insieme stabile E' SUBCLUSIVO.

L'algoritmo Greedy

Domanda: Se il problema di Ottimizzazione Combinatoria è subclusivo, l'algoritmo greedy fornisce SEMPRE una soluzione OTTIMA?

Consideriamo il problema di determinare il massimo insieme stabile su un grafo (tutti i nodi hanno peso pari ad 1).



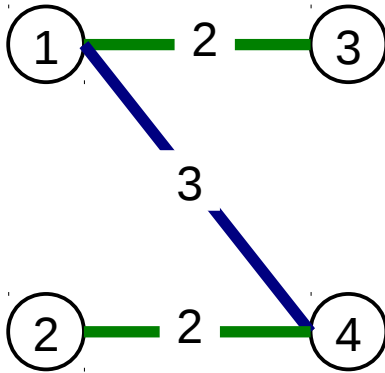
Soluzione greedy: $S^* = \{1,5\}$

Soluzione ottima: $S^* = \{3,4,6\}$

La soluzione trovata dall'algoritmo greedy è massimale ma non è ottima!

L'algoritmo Greedy

Consideriamo il problema di determinare il matching di peso massimo su un grafo bipartito.



Soluzione greedy: $S^* = \{(1,4)\}$

Soluzione ottima: $S^* = \{(1,3), (2,4)\}$

La soluzione trovata dall'algoritmo Greedy è massimale ma non ottima!

L'algoritmo greedy fallisce perché ogni volta trova una soluzione massimale che non è massima!

Una situazione del genere si verifica per tutti quei problemi di Ottimizzazione Combinatoria in cui F contiene insieme massimali di cardinalità diversa.

Proprietà di scambio

Domanda: Quali sono i problemi di Ottimizzazione Combinatoria in cui gli insiemi massimali in F hanno tutti stessa cardinalità?

Proprietà di scambio: Siano $F_1, F_2 \in F$, con $|F_1| < |F_2|$. F soddisfa la **proprietà di scambio** se è sempre possibile trovare un elemento $y \in F_2 - F_1$ tale che $F_1 \cup \{y\} \in F$.

Proposizione: Se F soddisfa la proprietà di scambio allora tutti i suoi insiemi massimali hanno stessa cardinalità.

Dim: Supponiamo che esistano due massimali F_1 e F_2 con $|F_2| > |F_1|$. Dalla proprietà di scambio è possibile trovare un elemento di F_2 che aggiunto a F_1 conserva l'ammissibilità dell'insieme. Ma allora F_1 non era massimale!

Matroidi

Teorema (Rado): Dato un problema di Ottimizzazione Combinatoria subclusivo l'algoritmo greedy determina una soluzione ottima qualunque sia la funzione peso c se e solo se F soddisfa la proprietà di scambio.

Definizione: Un *matroide* è individuato da una coppia (U, F) subclusiva che soddisfa la proprietà di scambio.

Siamo in grado di individuare dei problemi di Ottimizzazione Combinatoria che corrispondano a matroidi?

Matroide grafico

Dato un grafo non orientato $G = (V, E)$ siano:

- $U = E$
- $F = \{\text{sottoinsiemi } F \text{ di } U \text{ che non formano cicli}\}.$

Per ogni $F \in F$, $G_F = (V, F)$ è detto *foresta* di G .

Teorema: La coppia (U, F) è un matroide.

Dim: Sicuramente $\emptyset \in F$ e la famiglia F è subclusiva (ogni sottoinsieme di un insieme di archi privi di cicli è ancora privo di cicli).

Dimostriamo che è verificata la proprietà di scambio.

Siano $A, B \in F$ tali che $|B| > |A|$ e consideriamo le foreste indotte $G_A = (V, A)$ e $G_B = (V, B)$.

Matroide grafico

Dim: Osserviamo che una foresta è l'unione disgiunta di t alberi. In generale, una foresta $G_F = (V, F)$ contiene esattamente $t=|V|-|F|$ alberi.

Infatti, indichiamo con

- V_i = numero di vertici dell' i -esimo albero t_i
- E_i = numero di archi dell' i -esimo albero t_i

Ne segue che

$$|F| = \sum_{i=1}^t E_i = \sum_{i=1}^t (V_i - 1) = \sum_{i=1}^t V_i - t = |V| - t$$

Pertanto

- G_A contiene $|V| - |A|$ alberi
- G_B contiene $|V| - |B|$ alberi

Matroide grafico

Poiché $|B| > |A|$, $|V|-|B| < |V|-|A|$ e quindi G_A contiene un numero di alberi maggiore rispetto a G_B . Pertanto:

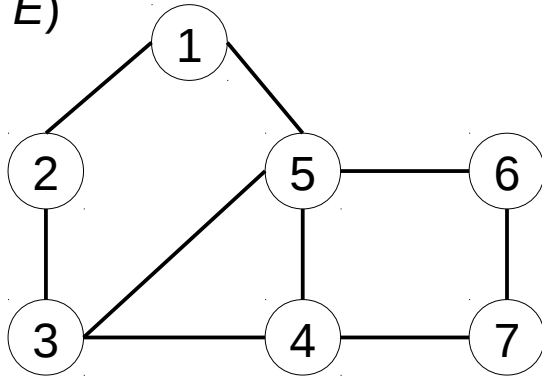
- Esiste un albero T in G_B i cui nodi appartengono ad alberi diversi di G_A .
- Poiché T è connesso, T deve contenere almeno un arco (u,v) i cui vertici sono in alberi diversi di G_A .

Dal momento che (u,v) connette vertici in due alberi diversi di G_A , sicuramente l'insieme $A \cup \{(u,v)\}$ non forma cicli e quindi appartiene ad F .

Pertanto la proprietà di scambio è verificata e è (U, F) un matroide.

Esempio

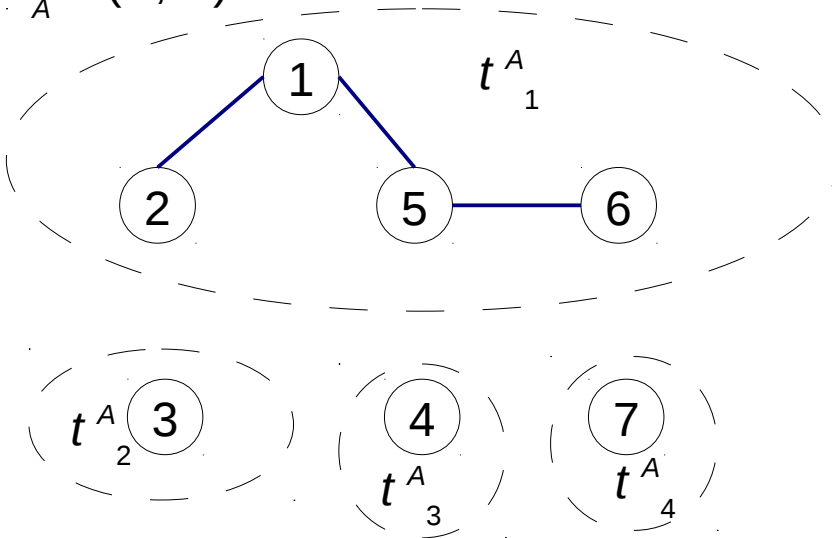
$G = (V, E)$



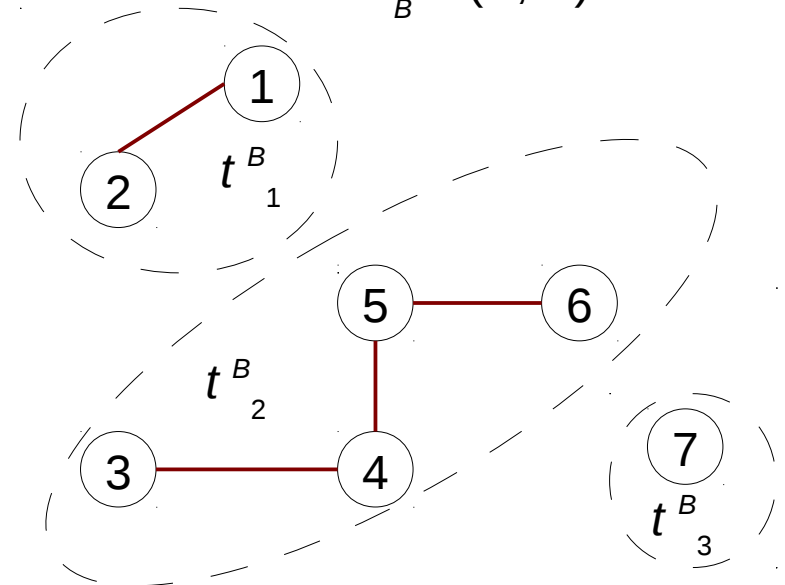
$A = \{12, 15, 56\}$

$B = \{12, 34, 55, 56\}$

$G_A = (V, A)$



$G_B = (V, B)$

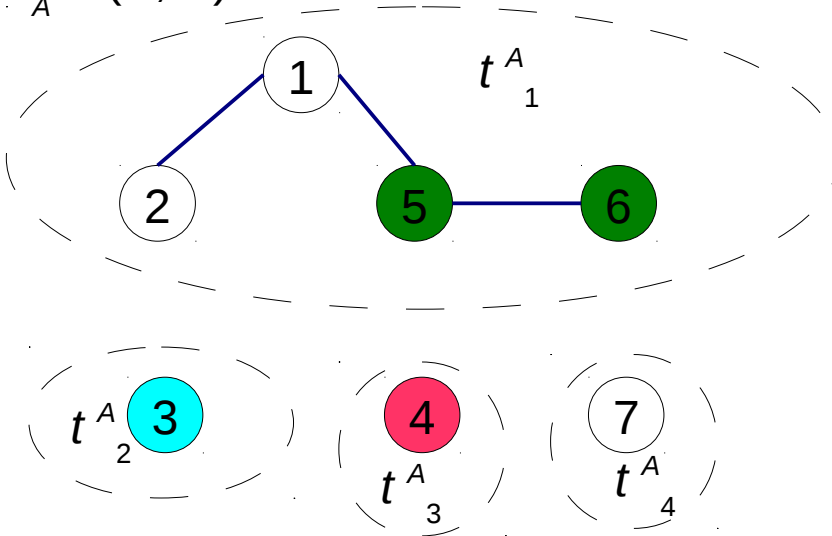


Esempio

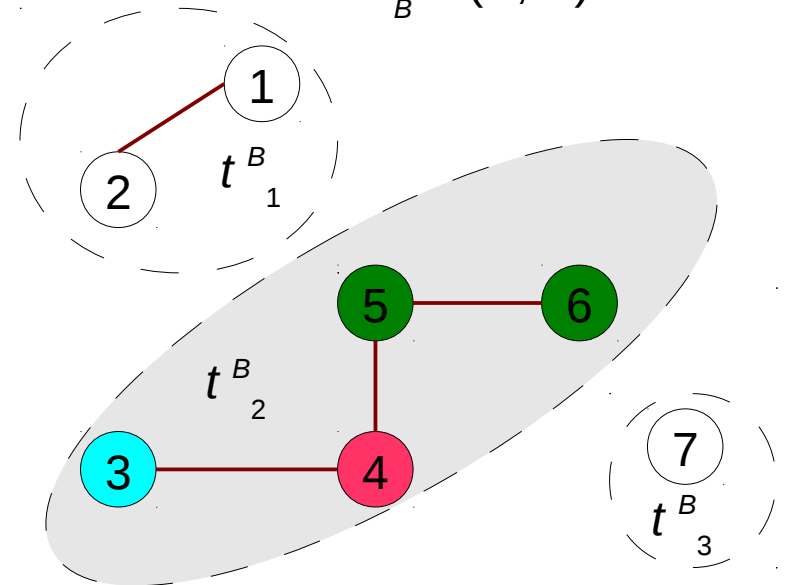
Consideriamo l'albero $T = t_2^B$. I nodi di T appartengono ad alberi diversi su G_A :

- $5, 6 \in t_1^A$
- $3 \in t_2^A$
- $4 \in t_3^A$

$G_A = (V, A)$



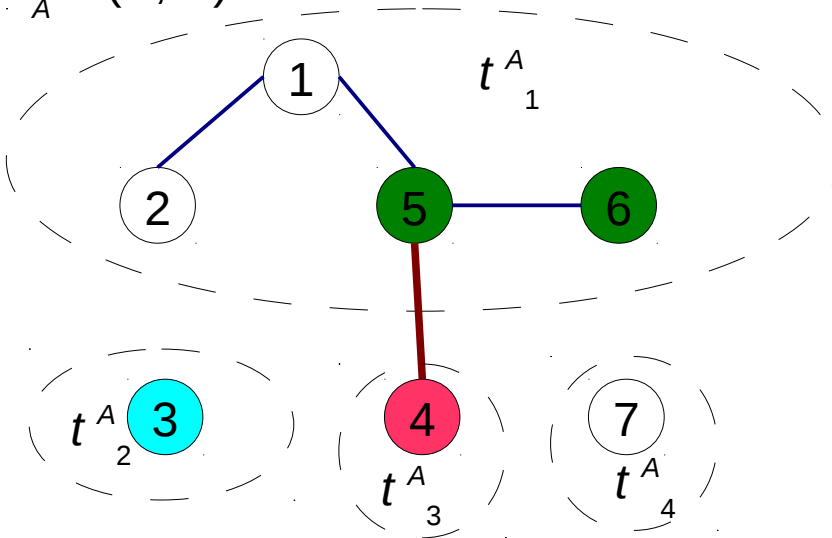
$G_B = (V, B)$



Esempio

Consideriamo l'arco 54 di T . I nodi di questo arco appartengono ad alberi diversi su G_A . Pertanto se aggiungiamo l'arco 54 ad A il grafo risultante è ancora una foresta.

$G_A = (V, A)$



$G_B = (V, B)$

