PARTE SECONDA:
RETI SENZA FILI
2.B. RETI DI SENSORI

IL PROBLEMA DEL DISPIEGAMENTO CENTRALIZZATO DI SENSORI MOBILI OVVERO L'ACCOPPIAMENTO PERFETTO DI PESO MINIMO

Prof. Tiziana Calamoneri

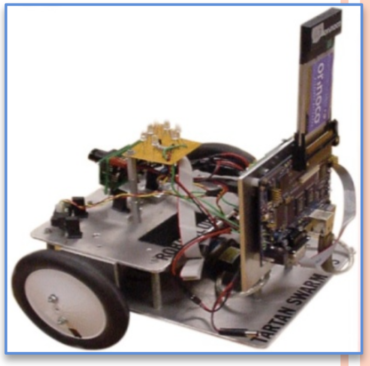
Corso di Algoritmi per le reti

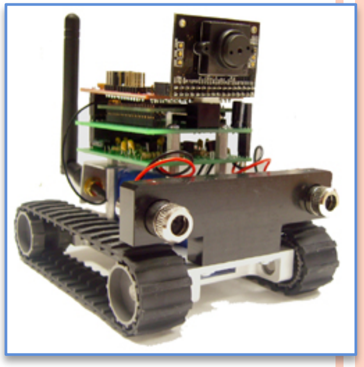
A.A. 2012/13

Il Problema

SENSORI MOBILI

- Dispositivi di piccola dimensione e basso costo (~150 $)
- Unità di monitoraggio (sensing)
- Unità rice/trasmissiva
- Piccola batteria
- Sistema di locomozione





Sono particolarmente utili in ambienti critici (ad esempio: in presenza di incendi, di esalazioni tossiche, di campi minati, ...)

Data un'area (AoI) da coprire:

IL PROBLEMA (1)

Si può pensare che ogni sensore monitorizzi un disco centrato nella sua posizione e di raggio *r*=raggio di sensing.

Lo scopo è raggiungere la copertura dell'AoI (stato di equilibrio o finale).

IL PROBLEMA (2)

Algoritmo di coordinamento dei dispositivi

Config. iniziale → Config. desiderata

può essere:
- casuale
- da una postazione sicura

può essere:
- tassellazione regolare
- qualunque altra cosa

purché l'area sia coperta

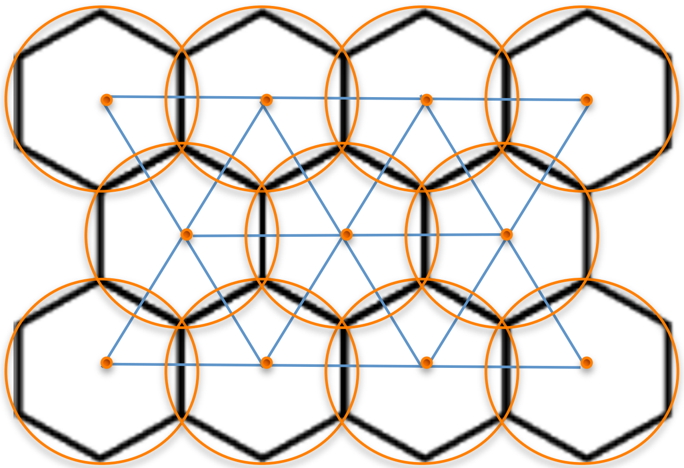
- Allo stesso tempo, vanno ottimizzati diversi parametri:
  - Distanza percorsa
  - Numero di mosse (start/stop)
  - Costo di comunicazione
  - Costo di computazione

Il problema (3)

- Distanza percorsa:
  - È il costo dominante
- Numero di mosse
  - Gli start/stop sono più costosi del movimento continuo
- Costo di comunicazione
  - Dipende dal numero di messaggi e dalla dimensione dei pacchetti ad ogni trasmissione
- Costo di computazione
  - Usualmente trascurabile, a meno che non si usino dei processori molto sofisticati

Il problema (4)

E' ben noto che la copertura ottima tramite cerchi tutti della stessa dimensione è quella che posiziona i centri in forma di griglia triangolare di dimensione opportuna ($\sqrt{2}$ r).



IL PROBLEMA (5)

Nel caso centralizzato:

- Vogliamo garantire la copertura assegnando ciascun sensore ad una posizione della griglia
- Vogliamo minimizzare l'energia consumata
- Trasformiamo nel classico problema dell'accoppiamento perfetto di peso minimo
- N.B. Funziona solo per una soluzione centralizzata, in cui sia nota l'AoI e la posizione iniziale di ciascun sensore

Il Modello del Problema su Grafi

Il Modello su Grafi (1)

- Riformuliamo il problema:
- Insieme di n sensori mobili $S=\{S_1, S_2, \ldots, S_n\}$
- Insieme di p locazioni sull'AoI $L=\{L_1, L_2, \ldots, L_p\}$
- $n \geq p$ (per garantire la copertura)
- Per ciascun S_i, determinare la locazione L_j che dovrà raggiungere, in modo da minimizzare l'energia totale spesa.

IL MODELLO SU GRAFI (2)

- Costruiamo un grafo bipartito pesato $G=(S \cup L, E, w)$ come segue:
  - per ogni sensore S_i si costruisce un nodo
  - per ogni punto del piano L_j si costruisce un nodo
  - si congiunge S_i con L_j tramite un arco per ogni i e j
  - si definisce la funzione peso $w(e_{ij})$ come un valore proporzionale al dispendio di energia necessario al sensore S_i per raggiungere la posizione L_j
  - si vuole scegliere l'accoppiamento di sensori/ posizioni che minimizzi il dispendio energetico

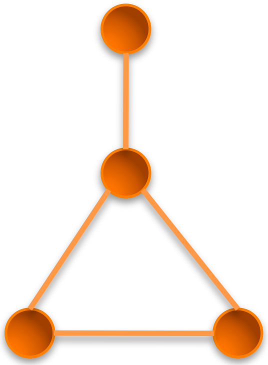
L'Accoppiamento Perfetto in Grafi Bipartiti

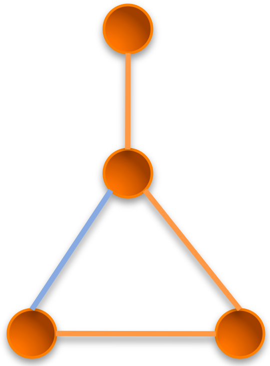
ACCOPPIAMENTO (1)

- **Def.** Un accoppiamento è un insieme di archi $M \subseteq E$ tale che nessun nodo è estremo di più di un arco di M.
- Accoppiamento Massimale
  - Non esiste alcun $e \notin E$ tale che $M \cup \{e\}$ sia un accoppiamento
- Accoppiamento Massimo
  - Accoppiamento con $|M|$ il più grande possibile
- Accoppiamento Perfetto
  - $|M| = n/2$: ogni nodo è estremo di un arco in M.

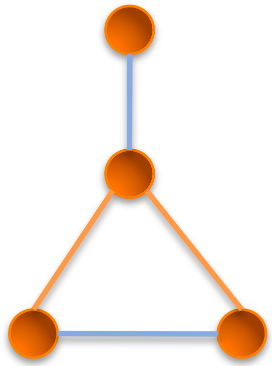
ACCOPPIAMENTO (2)

Esempio





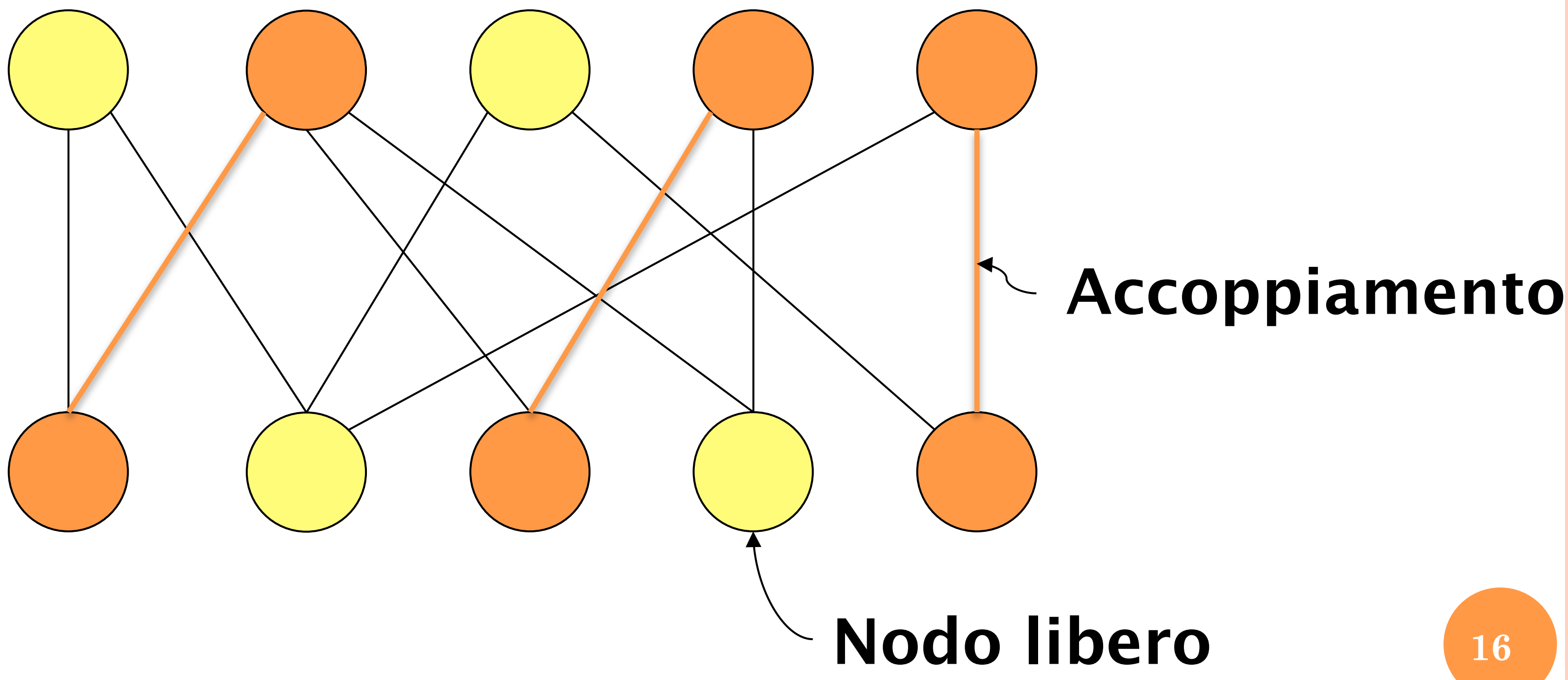
accoppiamento massimale



accoppiamento massimo

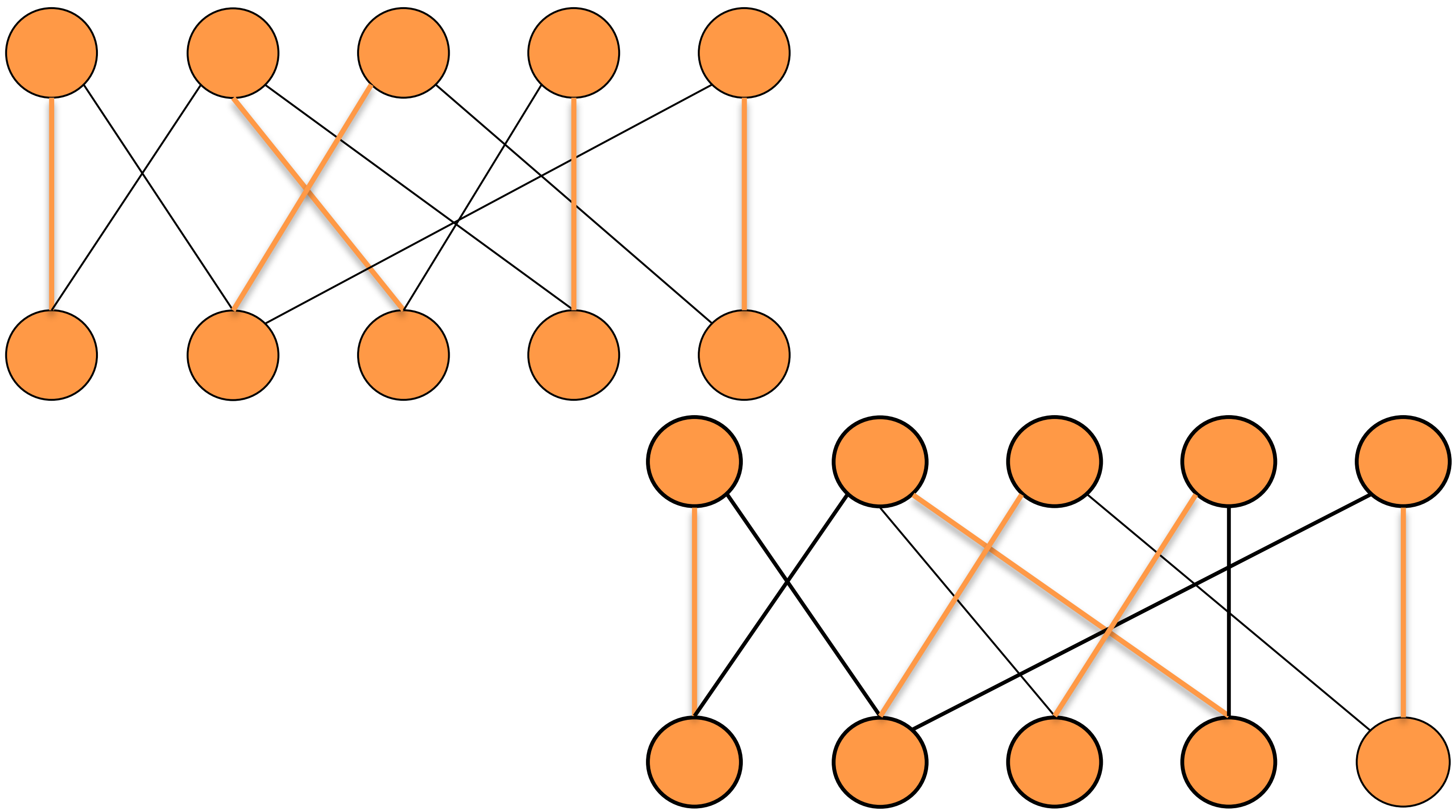
ACCOPPIAMENTO (3)

- Nomenclatura



ACCOPPIAMENTO (4)

- N.B. L'accoppiamento massimo non è unico



ACCOPPIAMENTO (5)

Problema originale: Problema dei matrimoni

- I nodi di un insieme rappresentano gli uomini
- I nodi dell'altro insieme le donne
- Un arco unisce una coppia che si piace

- L'accoppiamento massimo cerca di massimizzare il numero di coppie

Problemi di Accoppiamento

- Dato un grafo G, trovare un:
  - Accoppiamento Massimale: facile (greedy)
  - Accoppiamento Massimo
    - Tempo polinomiale; non facile.
    - Caso importante più facile: grafi bipartiti
  - Accoppiamento Perfetto
    - Caso particolare di accoppiamento massimo
    - Esistono teoremi appositi

Accoppiamento Massimo Bip. (1)

- **TH.** *(di P. Hall) G bipartito con* $|V_1| \leq |V_2|$, *allora G ha un accoppiamento perfetto sse per ogni insieme S di k nodi di* V_1 *vi sono almeno k nodi di* V_2 *adiacenti ad un nodo di S.*

  In altre parole, $\forall S \subseteq V_1$, $|S| \leq |adj(S)|$.
- **DIM.** Condizione necessaria: Se G ha un accoppiamento perfetto M ed S è un qualsiasi sottoinsieme di V_1, allora ogni nodo in S è accoppiato da M con un differente nodo in $adj(S)$, segue che $|S| \leq |adj(S)|$.

ACCOPPIAMENTO MASSIMO BIP. (2)

(segue dim. del th. di Hall) G bipartito con $|V_1| \leq |V_2|$*, allora G ha un accoppiamento perfetto sse* $\forall S \subseteq V_1$, $|S| \leq |adj(S)|$.

- **DIM.** Condizione sufficiente: Se è verificata la condizione di Hall esiste un accoppiamento perfetto. Per assurdo non esista, cioè sia M un accoppiamento massimo, con $|M|<|V_1|$; dimostriamo che esiste un accoppiamento M' con $|M'|=|M|+1$.

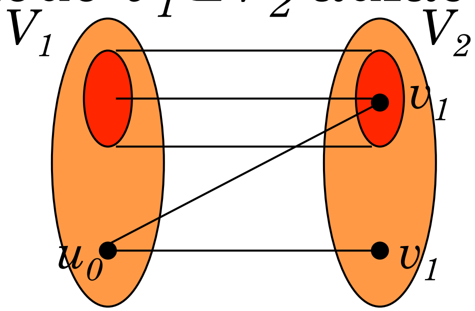
  Per ipotesi, $|M|<|V_1| \Rightarrow \exists\, u_0 \in V_1$ t.c. $u_0 \notin M$.

  Sia $S=\{u_0\}$ e vale che $1=|S| \leq |adj(S)|$ per ipotesi,

  e quindi esiste un nodo $v_1 \in V_2$ adiacente ad u_0.

  a. $v_1 \notin M$

  b. $v_1 \in M$



Accoppiamento Massimo Bip. (3)

(segue c.s.. del th. di Hall) G bipartito con $|V_1| \leq |V_2|$*, allora G ha un accoppiamento perfetto sse* $\forall S \subseteq V_1$, $|S| \leq |adj(S)|$.

a. Se $v_1 \notin M$ fine

b. Consideriamo l'accoppiato di v_1 *in M*, u_1.

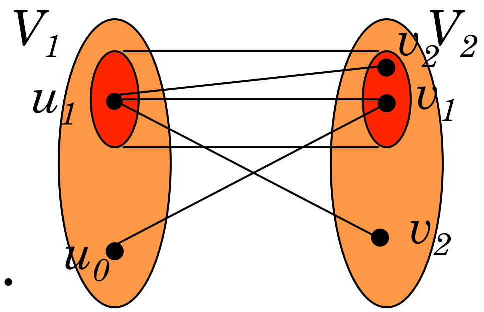
$S=\{u_0,u_1\}$ e $2=|S|\leq|adj(S)|$.

Deve esistere un altro nodo v_2,

distinto da v_1, e adiacente ad u_0 o ad u_1.

a. $v_2 \notin M$

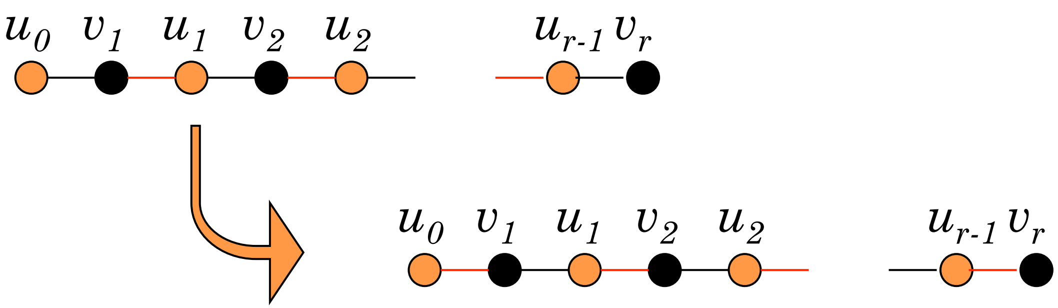
b. $v_2 \in M$



ACCOPPIAMENTO MASSIMO BIP. (4)

(segue c.s.. del th. di Hall) G bipartito con $|V_1|\leq|V_2|$, *allora G ha un accoppiamento perfetto sse* $\forall S\subseteq V_1$, $|S| \leq |adj(S)|$.

Continuando in questo modo, per la finitezza del grafo, si arriva necessariamente ad un nodo v_r che non appartiene ad M. Ognuno dei nodi v_i è adiacente ad almeno uno tra $u_0, u_1, \ldots, u_{i-1}$. Come nel caso r=2 si ha:



CVD

ACCOPPIAMENTO MASSIMO BIP. (5)

COR. *G bipartito k-regolare con* $|V_1|=|V_2|$, *allora G ha k accoppiamenti perfetti.*

DIM. Sia S un sottinsieme di V_1. $adj(S)$ ha al più $k|S|$ nodi (se ciascun nodo in $adj(S)$ ha grado 1 nel sottografo indotto da $S \cup adj(S)$) ed almeno $|S|$ nodi (se ciascun nodo in $adj(S)$ ha grado k nel sottografo indotto da $S \cup adj(S)$). In tutti i casi la condizione di Hall è verificata, e quindi esiste un accoppiamento perfetto, che può essere rimosso dal grafo dando luogo ad un nuovo grafo $(k-1)$-regolare. Per esso possiamo ripetere il ragionamento. **CVD**

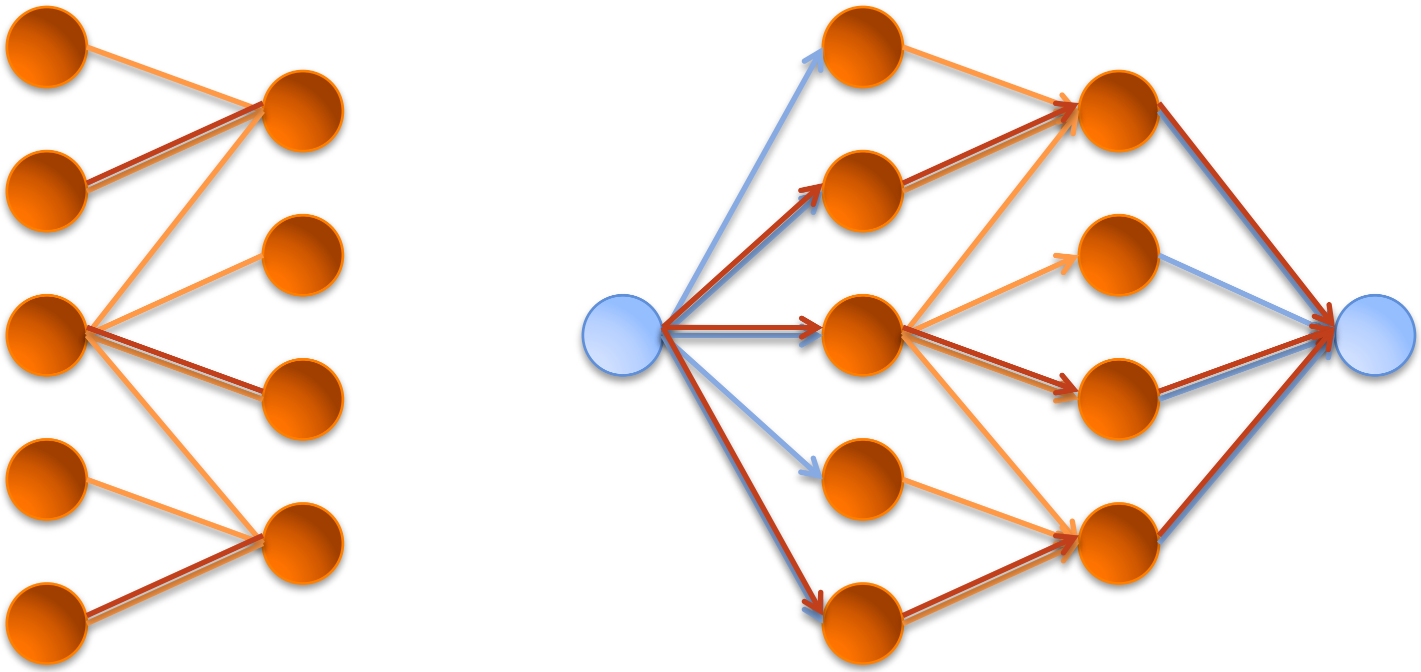
Accoppiamento Massimo Bip. (6)

- Il teorema di P. Hall non fornisce un metodo algoritmico per costruire un accoppiamento perfetto.
- Il problema dell'accoppiamento massimo in un grafo bipartito è equivalente al problema del massimo flusso in una rete.
- Dato $G=(V=V_1 \cup V_2, E)$, crea una rete di flusso $G'=(V', E')$ in cui i flussi corrispondono all'accoppiamento nel grafo originale:
  - Dalla sorgente s ai nodi in V_1 : $\{(s,u) \mid u \in V_1\}$
  - Da u in V_1 a v in V_2: $\{(u,v) \mid u \in V_1, v \in V_2, e\ (u,v) \in E\}$
  - Dai nodi in V_2 al pozzo t: $\{(v,t) \mid v \in V_2\}$
  - Capacità: $c(u,v) = 1$, for all $(u,v) \in E'$

ACCOPPIAMENTO MASSIMO BIP. (7)

- Fatto: Sia M un accoppiamento in un grafo bipartito G. Allora esiste un flusso f della rete G' t.c. $|M|=|f|$.

  Viceversa, se f è un flusso di G' allora esiste un accoppiamento M in G t.c. $|M|=|f|$.



ACCOPPIAMENTO MASSIMO BIP. (8)

- Th: (dell'integralità) *Se la capacità c assume solo valori interi, allora il flusso massimo f ha la proprietà che |f| è un valore intero. Inoltre, per tutti i nodi u e v, f(u,v) è un intero.*
- Corol.: *La cardinalità di un accoppiamento massimo M in un grafo bipartito G è uguale al valore di un flusso massimo f nella rete associata G'.*

Accoppiamento Massimo Bip. (9)

Dim. del Corol.: *La cardinalità di un accoppiamento massimo M in un grafo bipartito G è uguale al valore di un flusso massimo f nella rete associata G'.*

Sia M massimo e, per assurdo, f non sia massimo.

Allora esiste f' t.c. $|f'|>|f|$.

Per il th. dell'integralità, f' ha valori interi e quindi, per il fatto, gli corrisponde un accoppiamento M'.

$|M'|=|f'|>|f|=|M|$. Quindi M non è massimo.

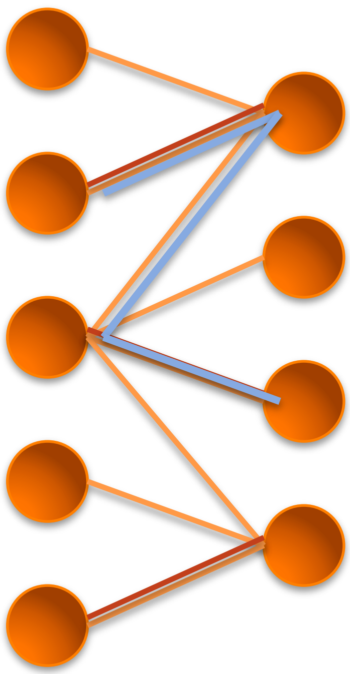
Analogamente, si dim. che se f è massimo allora anche M è massimo. **CVD**

ACCOPPIAMENTO MASSIMO BIP. (10)

- L'algoritmo di Ford-Fulkerson per il massimo flusso in una rete ha complessità $O(m|f|)$.
- Il flusso di G' ha al massimo cardinalità pari a $min\{|V_1|, |V_2|\}$. Quindi, la complessità di un eventuale algoritmo che sfrutta il flusso massimo è $O(nm)$.

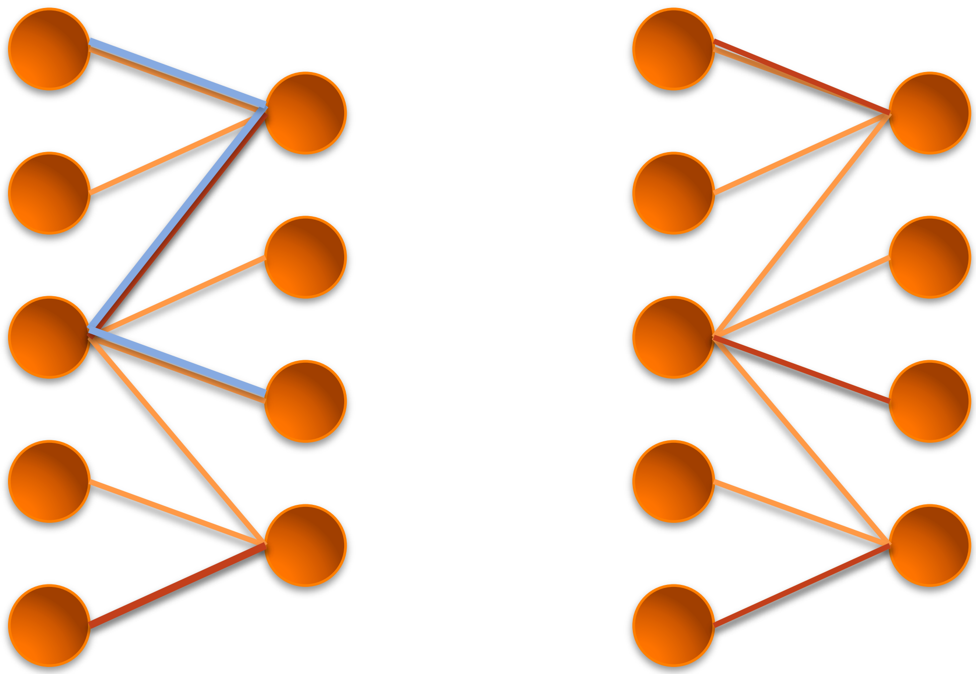
ACCOPPIAMENTO MASSIMO BIP. (11)

- Def. Dato un accoppiamento M in un grafo G, un cammino alternante rispetto ad M è un cammino che alterna archi dell'accoppiamento ad archi che non sono nell'accoppiamento.



ACCOPPIAMENTO MASSIMO BIP. (12)

- Def. Dato un accoppiamento M in un grafo G, un cammino aumentante rispetto ad M è un cammino alternante che inizia e termina in due nodi liberi dall'accoppiamento.



Dopo lo scambio degli archi dello accoppiamento con gli altri nel cammino aumentante, lo accoppiamento ha aumentato la sua cardinalità.

Accoppiamento Massimo Bip. (13)

- Th. (del Cammino Aumentante) [Berge 1975] *M è un accoppiamento massimo se e solo se non esiste alcun cammino aumentante rispetto ad M.*
- Dim. (➔) M massimo allora non ci sono cammini aumentanti. Negando, se ci sono cammini aumentanti, allora M non è massimo. Questo è ovvio, perché posso scambiare gli archi nel cammino ed accrescere M.

ACCOPPIAMENTO MASSIMO BIP. (14)

(Segue dim. del Th. *M è un accoppiamento massimo se e solo se non esiste alcun cammino aumentante rispetto ad M.)*

- Dim. (🡐) Non esistono camm. aumentanti, allora M massimo.

  Per assurdo M non è massimo. Sia M' t.c.

  $|M'|>|M|$.

  Si consideri il grafo H indotto da M ed M' dove gli archi che sono sia in M che in M' sono rappresentati 2 volte. Quindi H è un multigrafo.

ACCOPPIAMENTO MASSIMO BIP. (15)

Segue dim. Del Th. *M è un accoppiamento massimo se e solo se non esiste alcun cammino aumentante rispetto ad M.*

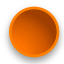
- *H* ha la proprietà:
  - Per ogni v in H, $deg(v) \leq 2$. (ogni nodo ha al più un arco di M ed uno di M')
- Segue che ogni componente connessa di H è un ciclo o un cammino.
  - cicli di lungh. pari, altrimenti esisterebbe un nodo estremo di due archi dello stesso accoppiamento (M o M'); assurdo per la def. di accoppiamento

Accoppiamento Massimo Bip. (16)

Segue dim. Del Th. *M è un accoppiamento massimo se e solo se non esiste alcun cammino aumentante rispetto ad M.*

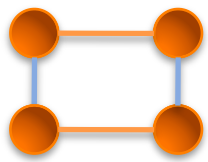
- Più nel dettaglio, le componenti connesse di H possono essere di 6 tipi:

1. un nodo isolato



2. un 2-ciclo

3. un 2k-ciclo, k>1



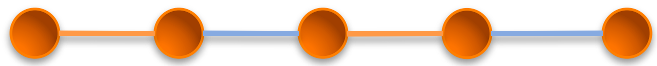
…

ACCOPPIAMENTO MASSIMO BIP. (17)

Segue dim. Del Th. *M è un accoppiamento massimo se e solo se non esiste alcun cammino aumentante rispetto ad M.*

…

4. un 2k-cammino



5. un (2k+1)-cammino i cui estremi sono incidenti ad *M*

6. un (2k+1)-cammino i cui estremi sono incidenti ad *M'*

Accoppiamento Massimo Bip. (18)

Segue dim. Del Th. *M è un accoppiamento massimo se e solo se non esiste alcun cammino aumentante rispetto ad M.*

- Oss. Ricordiamo che $|M|<|M'|$ per hp.
- Di tutte le componenti ora definite, solo 5 e 6 hanno un diverso num. di archi, e solo 6 ha più archi di M' che di M.
- Segue che deve esistere almeno una componente di tipo 6

- Tale comp. è un cammino aumentante per M. Assurdo. **CVD**

Accoppiamento Massimo Bip. (19)

- Utilizziamo il teorema del Cammino Aumentante per produrre un algoritmo iterativo che, ad ogni iterazione, cerca un nuovo cammino aumentante tramite una modifica di una visita in ampiezza partendo dai nodi che l'accoppiamento non tocca. In questo modo, i nodi sono strutturati in livelli (toccati e non toccati dall'accoppiamento).

ACCOPPIAMENTO MASSIMO BIP. (20)

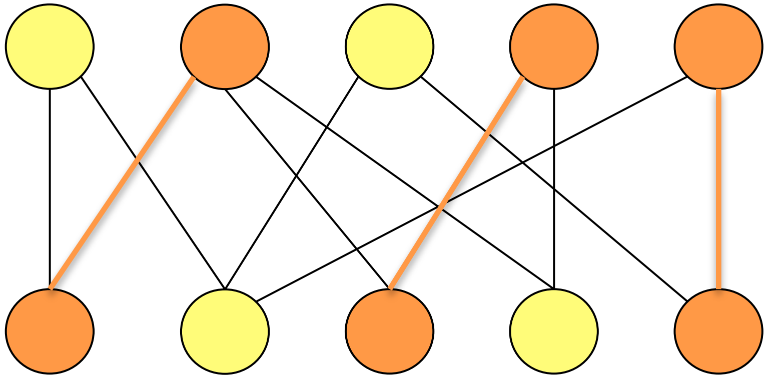
Idea dell'algoritmo:

- Parti da un accoppiamento arbitrario (anche vuoto)
- Finché esistono cammini aumentanti:
  - Trova il cammino aumentante P
  - Scambia in P gli archi dell'accoppiamento con gli altri

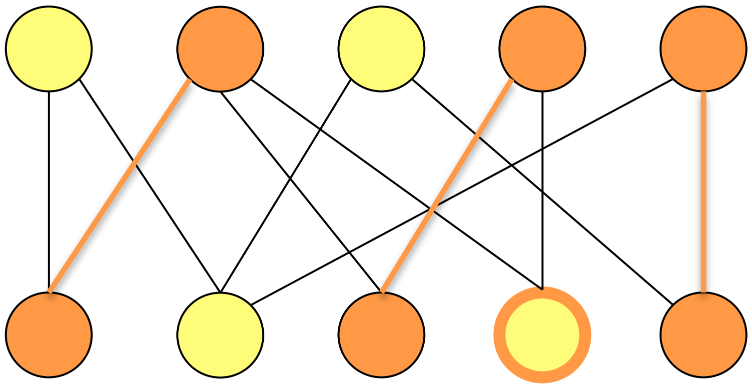
Complessità: dipende dalla complessità di cercare il cammino aumentante.

ACCOPPIAMENTO MASSIMO BIP. (21)

- Parti da un accoppiamento arbitrario

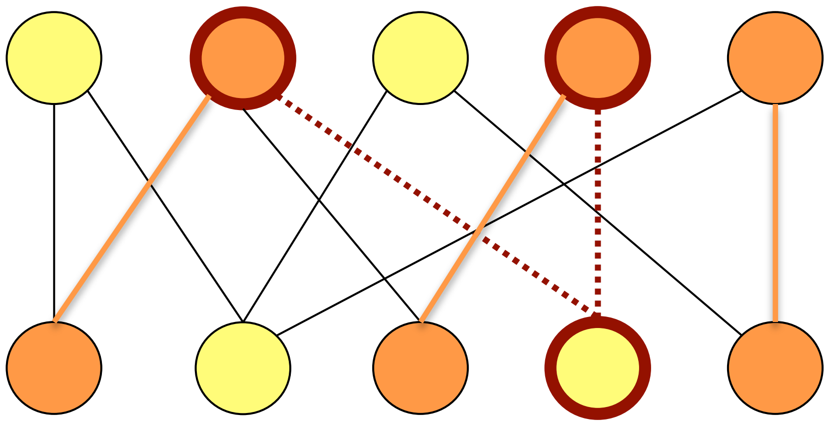


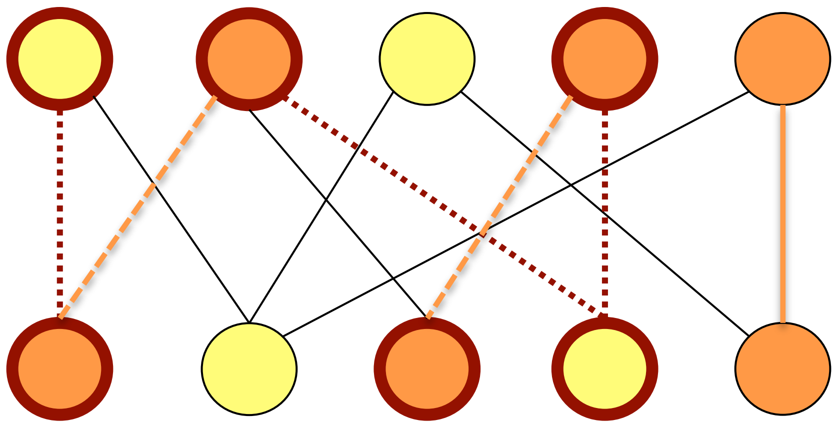
- Scegli un nodo libero…



- E visita i suoi adiacenti…

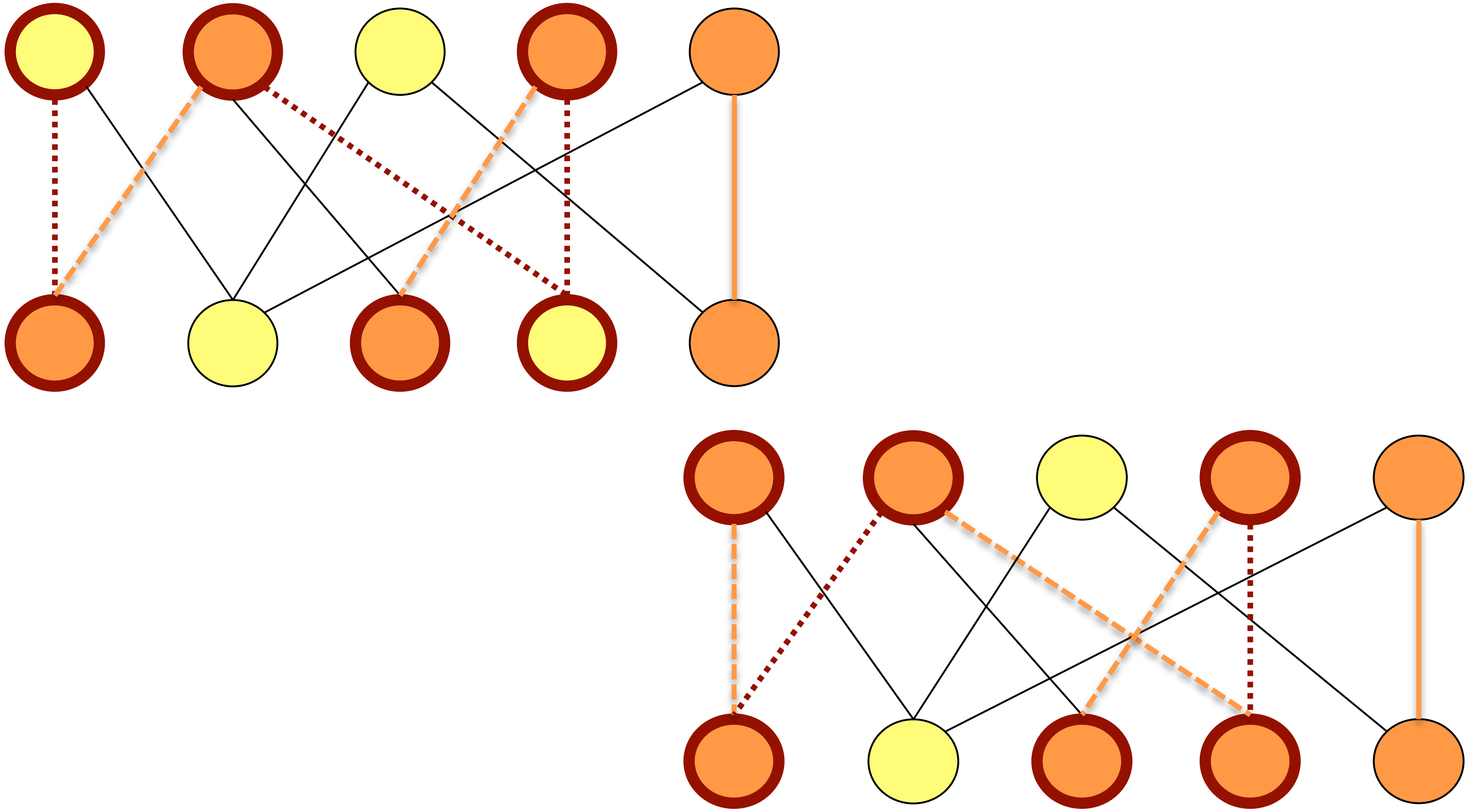
Accoppiamento Massimo Bip. (22)





… prosegui finché non viene raggiunto un altro nodo libero, e quindi si trova un cammino aumentante

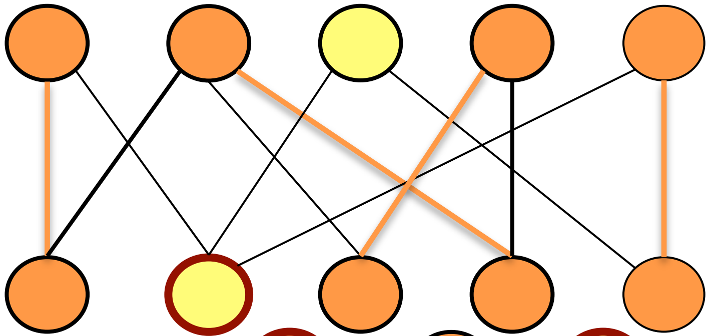
ACCOPPIAMENTO MASSIMO BIP. (23)

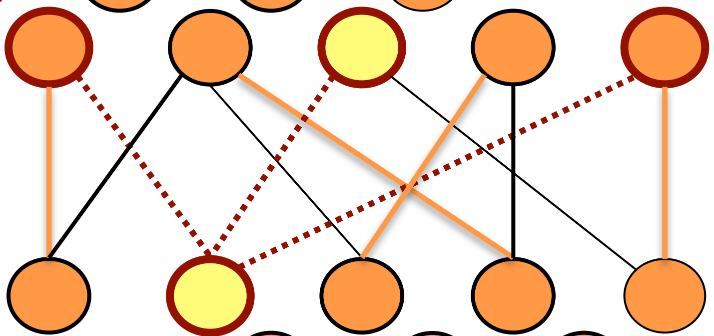


Scambia il ruolo degli archi dell'accoppiamento e degli altri

Accoppiamento Massimo Bip. (24)

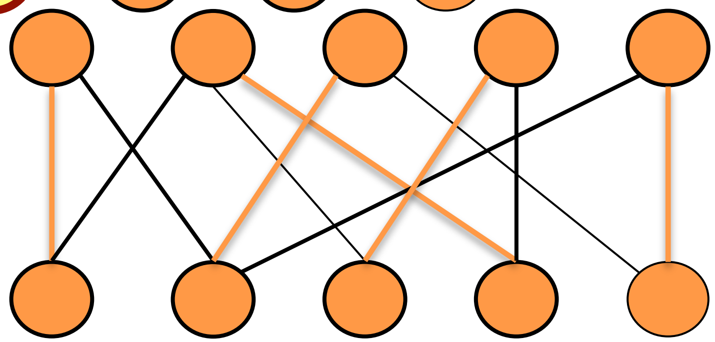
Ripeti: prendi un altro nodo libero…





…visita i suoi adiacenti e gli adiacenti degli adiacenti…

… e scambia



Non ci sono altri cammini aumentanti: Fine

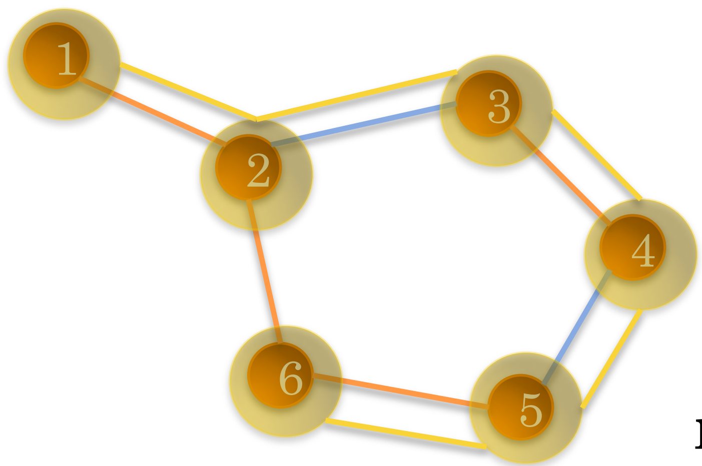
ACCOPPIAMENTO MASSIMO BIP. (25)

- Problema: come si trova un cammino aumentante per M?
- Idea:
  - Parti da un nodo libero
  - Esegui una DFS modificata come segue:
    - tieni traccia del livello corrente
    - se il livello è pari, usa un arco di M
    - se il livello è dispari, usa un arco di E-M
    - Appena trovi un nodo libero hai trovato un cammino aumentante

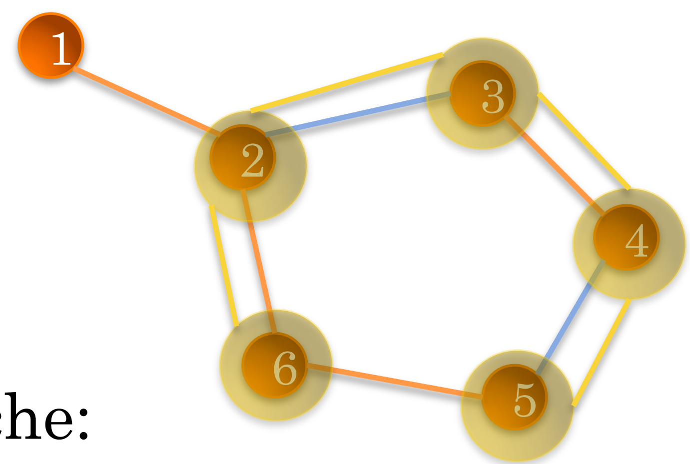
Accoppiamento Massimo Bip. (26)

Esempio:

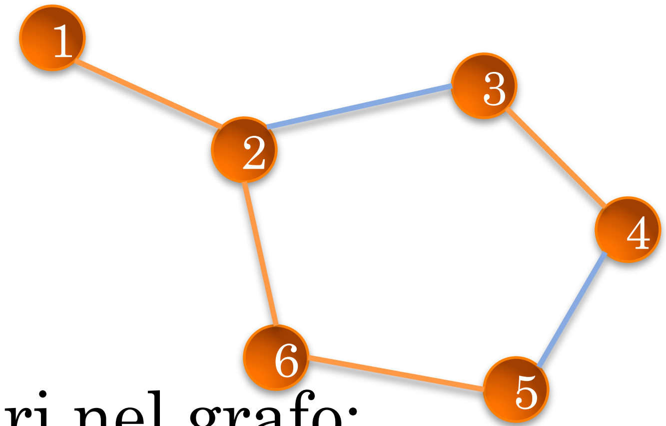
- Parti da un nodo libero
- Esegui una DFS modificata come segue:
  - tieni traccia del livello corrente
  - se il livello è pari, usa un arco di M
  - se il livello è dispari, usa un arco di E-M
  - Appena trovi un nodo libero hai trovato un cammino aumentante



ma anche:



ACCOPPIAMENTO MASSIMO BIP. (27)



- Problema: presenza di cicli dispari nel grafo:
  - in un ciclo dispari c'è sempre un nodo libero con due archi non in M che contribuiscono al ciclo
  - se la DFS percorre il ciclo nella direzione "sbagliata" il cammino aumentante non viene trovato
- Grafi senza cicli dispari: grafi bipartiti

Accoppiamento Massimo Bip. (28)

Algoritmo TrovaCamminoAumentante $(G=(U \cup W,E), M)$

- parti da un nodo libero di U
- se il nodo corrente è in U segui un arco non in M
- altrimenti segui un arco in M
- appena trovi un nodo libero di W hai trovato un cammino aumentante

Complessità: $O(n+m)$

Complessità dell'algoritmo che trova l'accoppiamento massimo: $n/2[O(n+m)+O(n)]=O(nm)$

max n. di iterazioni

inversione archi del cammmino aum.

ACCOPPIAMENTO MASSIMO BIP. (29)

- L'algoritmo di Hopcroft–Karp (1973) trova un accoppiamento massimo di un grafo bipartito in tempo $O(m\sqrt{n})$.
- L'idea è analoga alla precedente, e consiste nell'accrescere ripetutamente la cardinalità dell'accoppiamento parziale cercando cammini aumentanti.
- Invece di trovare un cammino aumentante per ogni iterazione, l'algoritmo cerca un insieme massimale di cammini aumentanti.
- In questo modo sono necessarie solo $O(\sqrt{n})$ iterazioni.

Accoppiamento Massimo Bip. (30)

Algoritmo di Hopcroft–Karp

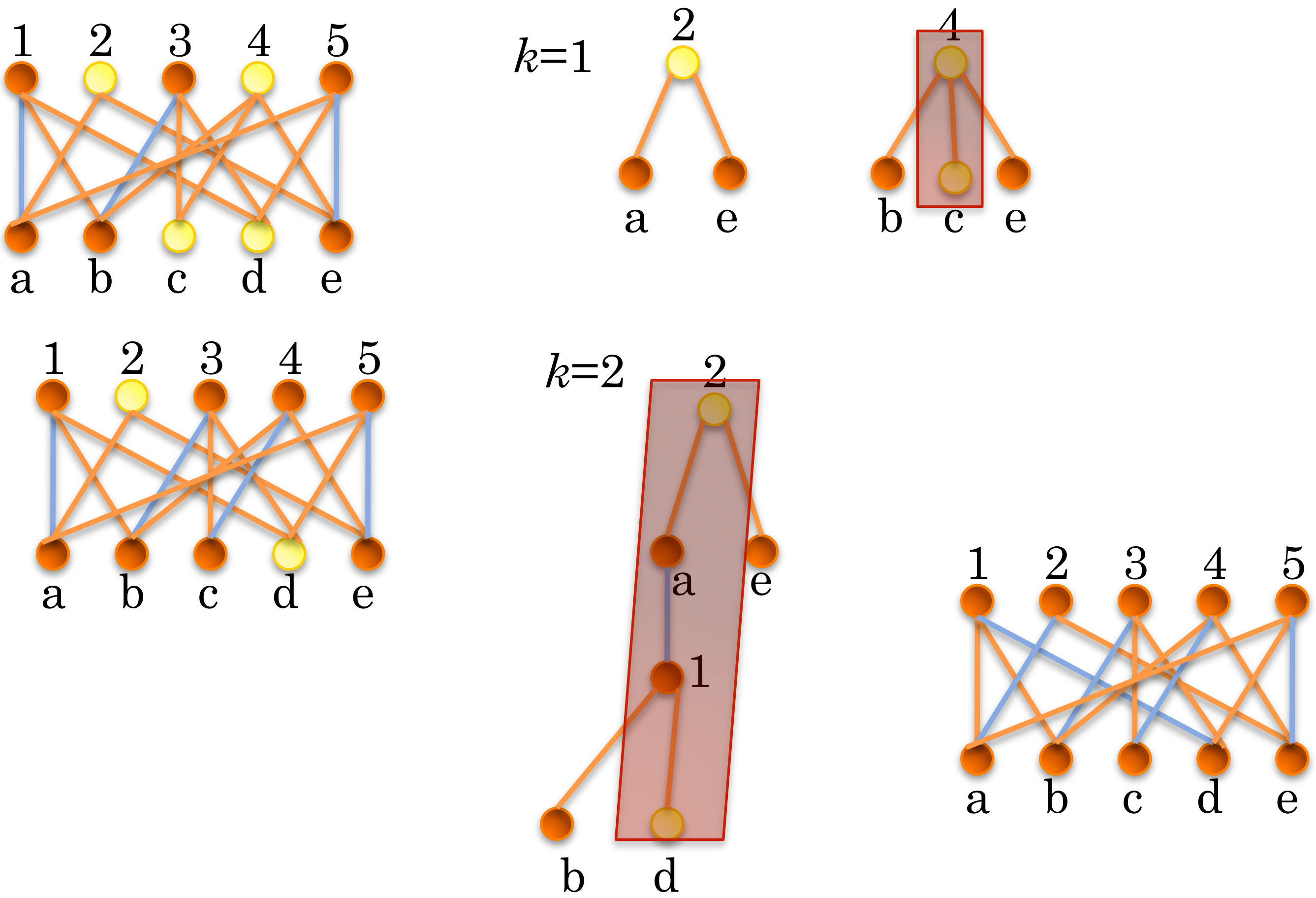
Passi della k-esima fase:

- breadth first search modificata partendo da tutti i nodi liberi di V_1. La visita termina quando vengono raggiunti nodi liberi di V_2 (al livello k).
- Tutti i nodi liberi di liv. k di V_2 sono messi in un insieme F.

  N.B. v è in F sse è la fine di un cammino aumentante
- Trova un insieme massimale di cammini aumentanti *vertex disjoint* di lungh. k usando una depth first search da F verso i nodi di partenza di V_1 (risalita di padre in padre).
- Ogni cammino trovato è un cammino aumentante usato per aumentare M.
- L'alg. termina quando non ci sono più cammini aumentanti trovati dal primo passo.

ACCOPPIAMENTO MASSIMO BIP. (31)

Esempio Algoritmo di Hopcroft–Karp



Accoppiamento Massimo Bip. (31)

Analisi dell'Algoritmo di Hopcroft–Karp (sketch)

- Ogni passo consiste di una breadth first search ed una depth first search. Perciò può essere implementato in $O(n+m)=O(m)$.
- I primi $\sqrt{n}$ passi prendono tempo $O(m \sqrt{n})$.
- N.B. ad ogni passo la lunghezza dei cammini aumentanti trovati è sempre maggiore poiché ad ogni passo k vengono trovati tutti i cammini di lungh. k e i rimanenti hanno lungh. maggiore.
- Dopo i primi $\sqrt{n}$ passi, il più corto cammino aumentante è lungo almeno $\sqrt{n}$.
- …

Accoppiamento Massimo Bip. (32)

Segue Analisi dell'Algoritmo di Hopcroft–Karp (sketch)

- La differenza simmetrica tra un possibile accoppiamento ottimo e l'accoppiamento parziale M trovato dai primi $\sqrt{n}$ passi è un insieme di cicli alternanti e di cammini aumentanti *vertex-disjoint*.
- Ciascuno di questi cammini ha lunghezza almeno $\sqrt{n}$, quindi ce ne possono essere al più $\sqrt{n}$ e la dim. dell'accoppiamento massimo è più grande di al più $\sqrt{n}$ archi rispetto ad M.
- Ogni passo dell'algoritmo aumenta M di almeno uno, quindi al più $\sqrt{n}$ ulteriori passi sono sufficienti.
- L'algoritmo esegue quindi al più $2\sqrt{n}$ passi, quindi prende tempo $O(m \sqrt{n})$ nel caso peggiore.

ACCOPPIAMENTO MASSIMO BIP. (33)

- In molti casi questa complessità può essere migliorata.
- Per esempio, nel caso medio di grafi sparsi bipartiti random, nel 2006 [Bast et al.] è stato provato che i cammini aumentanti hanno lunghezza logaritmica.
- Come conseguenza, l'alg. di Hopcroft–Karp necessita di $O(\log n)$ passi e quindi $O(m \log n)$ tempo totale.

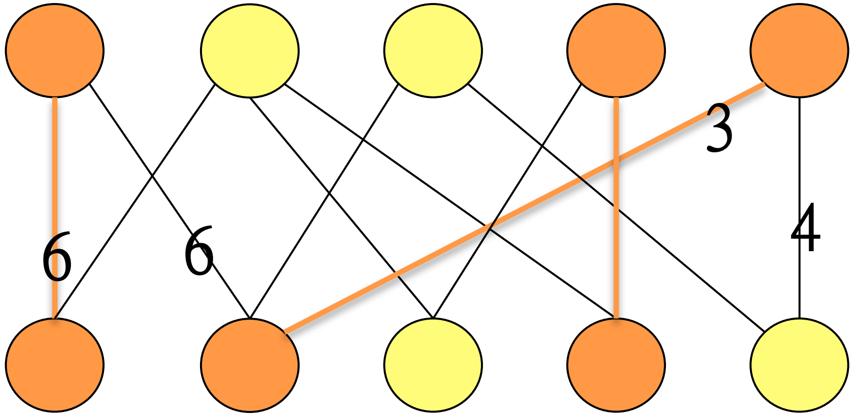
 

L’Accoppiamento Perfetto di Peso Minimo in Grafi Bipartiti

Accoppiamento Pesato (1)

- Ogni arco ha un costo
- Definizione di accoppiamento pesato: come prima (il peso non ha influenza se non si vuole ottimizzare alcuna funzione)
- Noi cerchiamo un accoppiamento perfetto di costo minimo
- N.B. Questo è equivalente a cercare un accoppiamento con costo massimo, in cui i pesi sono negativi

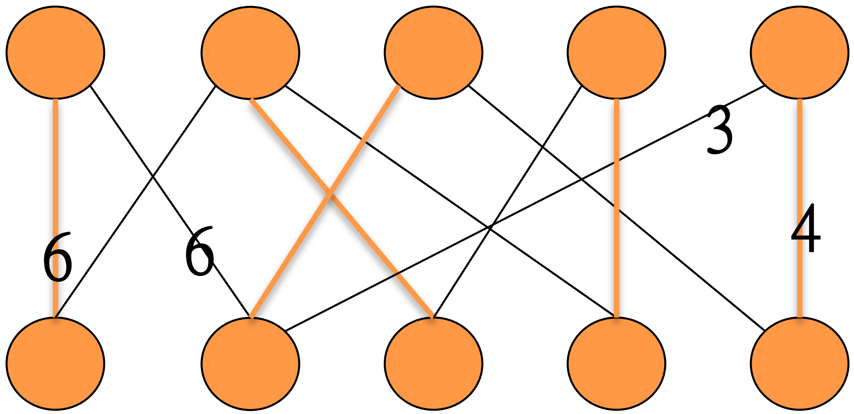
ACCOPPIAMENTO PESATO (2)



(gli archi senza peso hanno peso 1)

Costo dell'accoppiamento:
6+3+1=10

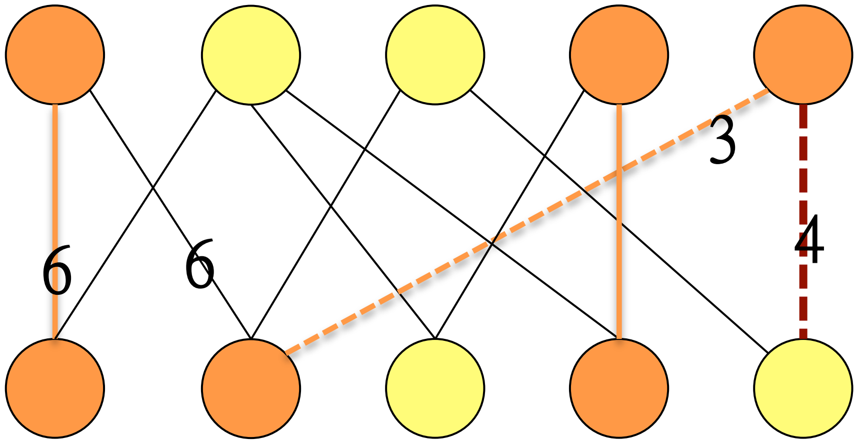
Accoppiamento di costo massimo:
6+4+1+1+1=13



ACCOPPIAMENTO PESATO (3)

Def. Cammino aumentante (diversa da prima!) Ogni cammino alternante tale che il costo totale degli archi non accoppiati > del costo degli archi accoppiati.

Costo del cammino aumentante= Costo degli archi non accoppiati-costo degli archi accoppiati



N.B. Ora i cammini aumentanti non devono finire necessariamente con un arco fuori dell'accoppiamento.

Accoppiamento Pesato (4)

Algoritmo:

- Inizia con un accoppiamento vuoto
- Ripeti
  - Trova un cammino aumentante P con costo massimo
  - Se il costo > 0, scambia il ruolo degli archi
  - Altrimenti ritorna l'accoppiamento di massimo peso.

- Complessità: almeno $O(nm)$.

ACCOPPIAMENTO PESATO (5)

- E' possibile definire il problema dell'accoppiamento di minimo peso come un problema di programmazione lineare (Hungarian method):
  - Dato un accoppiamento M, sia x la sua matrice di incidenza, dove $x_{ij} = 1$ se (i, j) è in M e 0 altrimenti.
  - Il problema diviene:

    minimizzare $\sum_{i,j} c_{ij} x_{ij}$ soggetto a

$$\sum_j x_{ij} = 1, i \in A$$

$$\sum_i x_{ij} = 1, j \in B$$

$$x_{ij} \geq 0, i \in A, j \in B$$

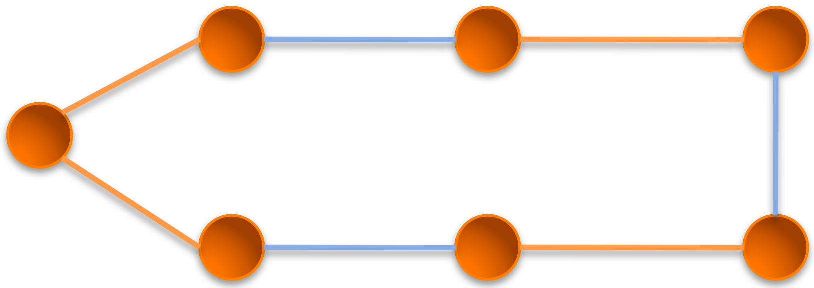
$$x_{ij} \text{ integer}, i \in A, j \in B$$

- Complessità: $O(n^3)$.

L'Accoppiamento Massimo in Grafi Qualunque

BOCCIOLI (BLOSSOMS) (1)

- Abbiamo detto che il problema dei grafi qualunque risiede nei cicli dispari contenenti un numero massimale di archi dell'accoppiamento



- Tali cicli sono detti boccioli (blossoms)

BOCCIOLI (BLOSSOMS) (2)

- Lemma (della contrazione dei cicli). Sia M un accoppiamento di G e B un bocciolo. Sia B nodo-disgiunto dal resto di M. Sia G' il grafo ottenuto da G contraendo B in un singolo nodo. Allora M' di G' indotto da M è massimo in G' sse M è massimo in G.
- Dim. M max in G => M' max in G'

  P.A. M' non è max. Quindi esiste un cammino aumentante P in G' rispetto ad M'. Sia b il nodo che rappresenta B.

  Due casi:

1. il cammino non passa per b => P aumentante anche per M. ASSURDO

BOCCIOLI (BLOSSOMS) (3)

Segue dim. del Lemma della contrazione dei cicli

2. il cammino passa per b => b è un estremo di P poiché gli archi di G' incidenti a b non sono in M per hp.

   Sia v il nodo libero di B

   Definisci $P' = P \cup P''$ dove P'' è dentro B e congiunge b con v.

   P' aumentante per G. ASSURDO.

BOCCIOLI (BLOSSOMS) (4)

Segue dim. del Lemma della contrazione dei cicli

- M' max in G' => M max in G

  $P.A.$ M non è max. Sia P un cammino aumentante in G per M.

  Due casi:

1. P non passa per b => P aumentante per G'. ASSURDO
2. P passa per b. Poiché B contiene un solo nodo libero, almeno un estremo di P è fuori di B. Sia w.

   Sia P' il sottocammino di P che congiunge w con b.

   P' è un cammino aumentante per G'.

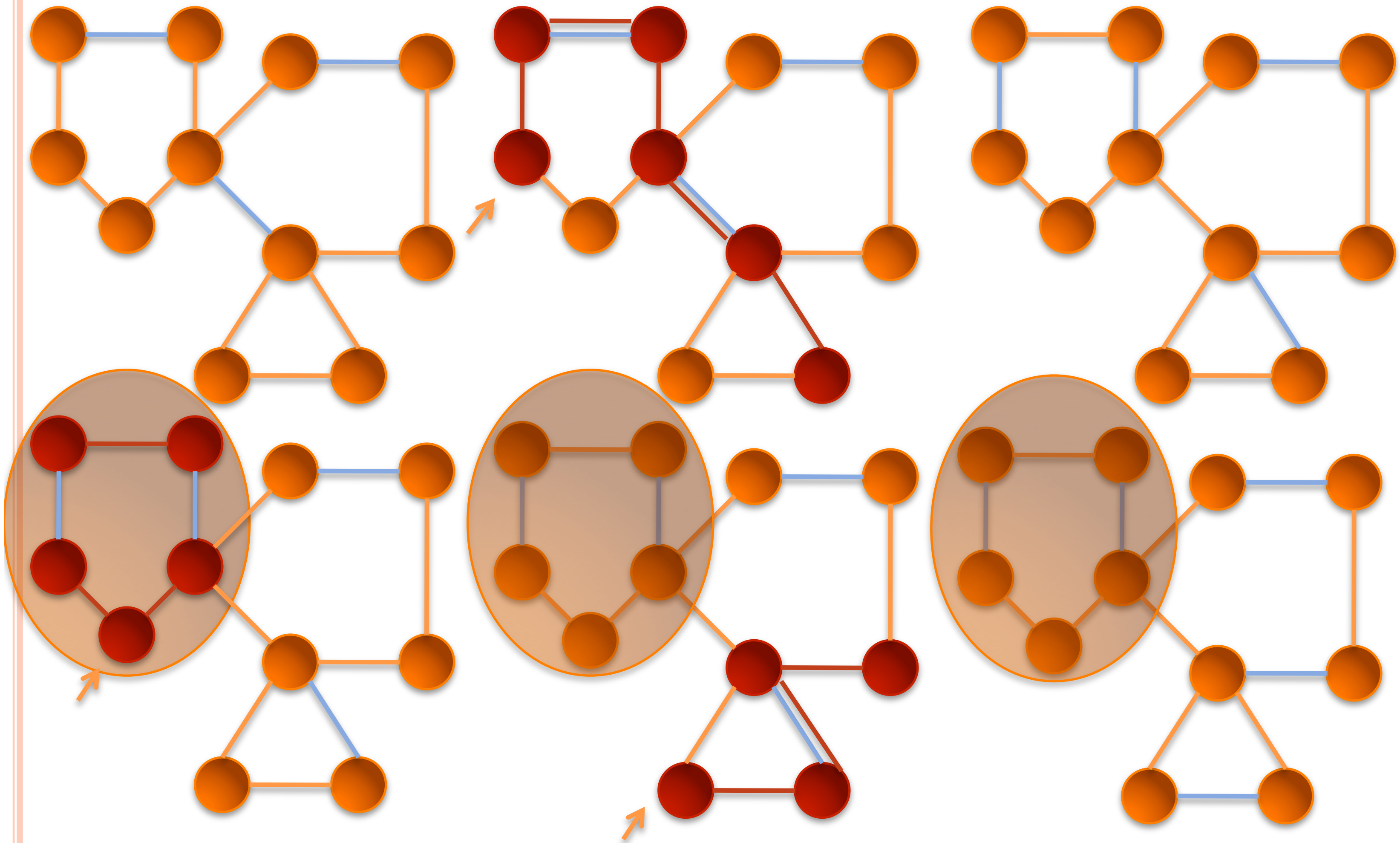
   ASSURDO. CVD

Accoppiamento Max in Grafi Qualunque (1)

- Per trovare un cammino aumentante in un grafo generale, “basta” modificare l’algoritmo per i bipartiti in modo che trovi anche i boccioli.
- Per ogni bocciolo trovato, questo viene contratto in un nodo e generato un nuovo grafo.
- Ogni cammino aumentante trovato sul nuovo grafo si traduce facilmente in un cammino aumentante in G.
- Per il lemma precedente, se M è massimo nel nuovo grafo, esso è massimo anche in G.

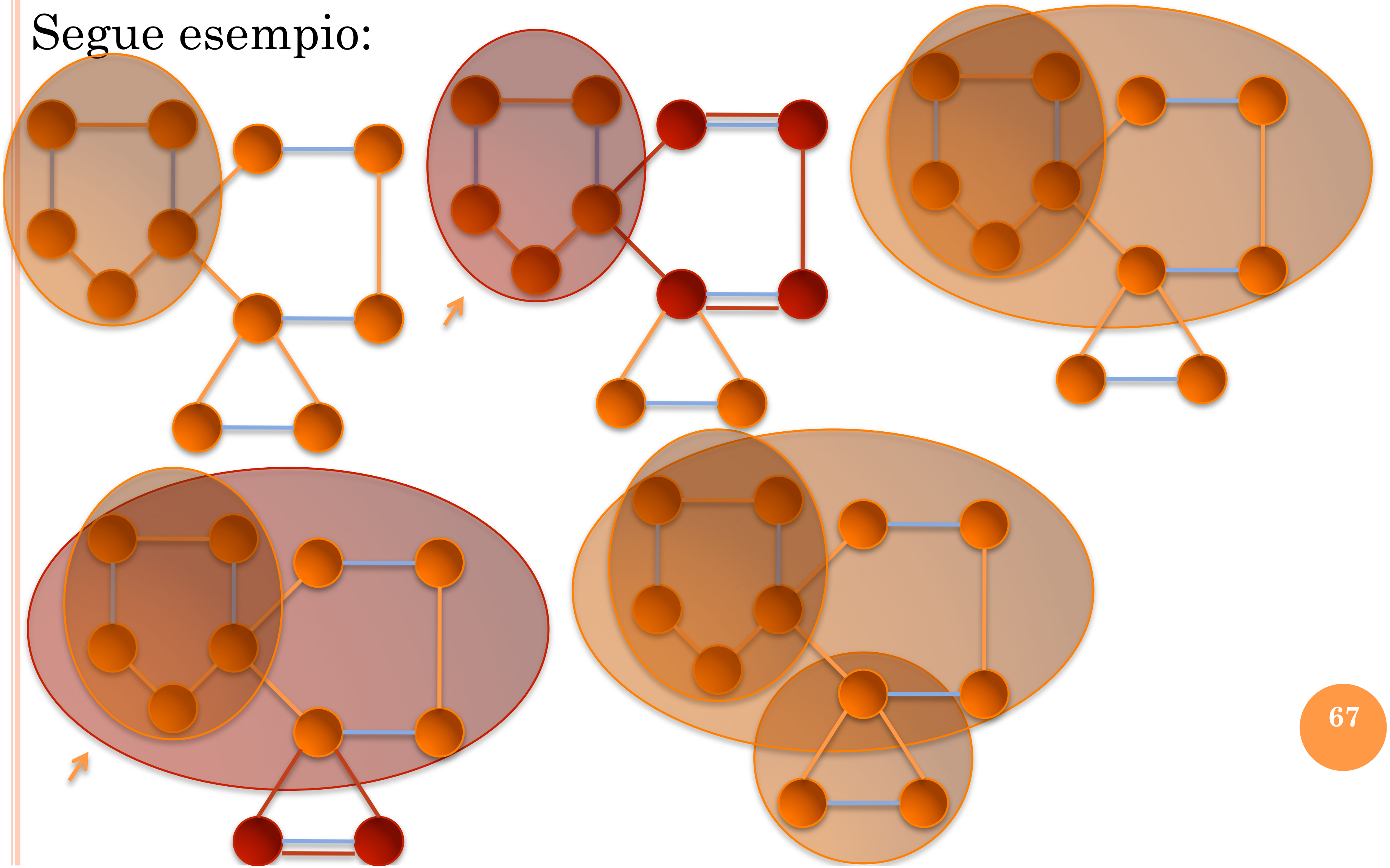
Accoppiamento Max in Grafi Qualunque (2)

Esempio:



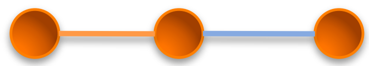
ACCOPPIAMENTO MAX IN GRAFI QUALUNQUE (3)

Segue esempio:



ACCOPPIAMENTO MAX IN GRAFI QUALUNQUE (4)

Algoritmo di Edmonds ['65]

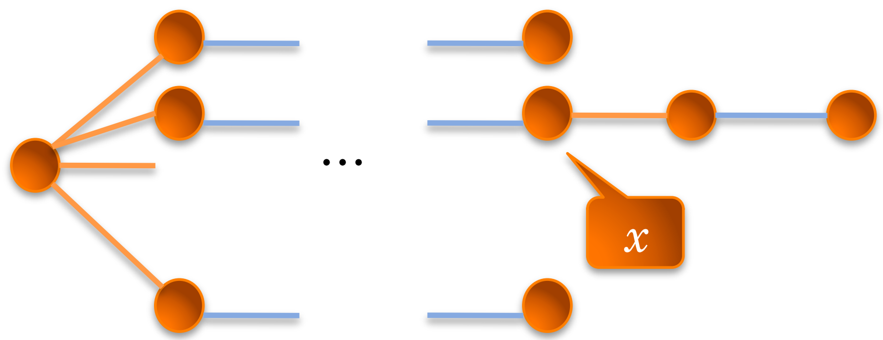
- M accoppiamento di G
- L sottinsieme dei nodi liberi (se L è vuoto => M max)
- F foresta t.c. ogni nodo di L corrisponde ad una componente di F
- Estendi F aggiungendo 
- Quindi: nodi a dist. dispari da elementi di L hanno grado 2 (1 in M e 1 in E-M): siano interni
- Gli altri nodi: esterni

(segue)

Accoppiamento Max in Grafi Qualunque (5)

Algoritmo di Edmonds – segue

- Considera i vicini dei nodi esterni.
- 4 possibilità:
    1. esiste x esterno incidente ad un y non in F:
       aggiungi ad F gli archi (x,y) ed (y,z), con (y,z) arco di M.



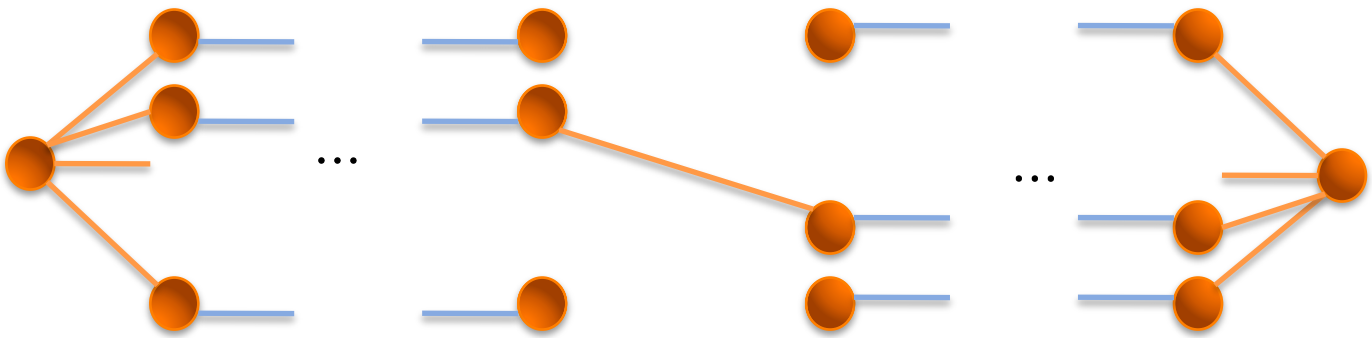
(segue)

Accoppiamento Max in Grafi Qualunque (6)

Algoritmo di Edmonds – segue

2. due nodi esterni in due diverse componenti di *F* sono adiacenti:

   cammino aumentante



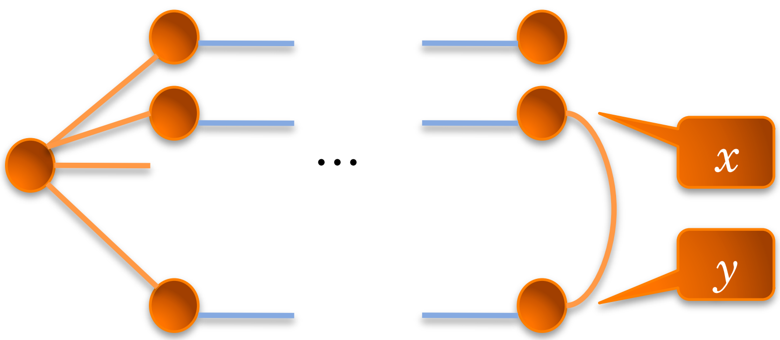
(segue)

Accoppiamento Max in Grafi Qualunque (7)

Algoritmo di Edmonds – segue

3. 2 nodi esterni x, y nella stessa componente di F sono adiacenti:

   sia C il ciclo che si forma. E' possibile spostare gli archi di M in C in modo che soddisfi la condizione del lemma della contrazione dei cicli => grafo ridotto G'



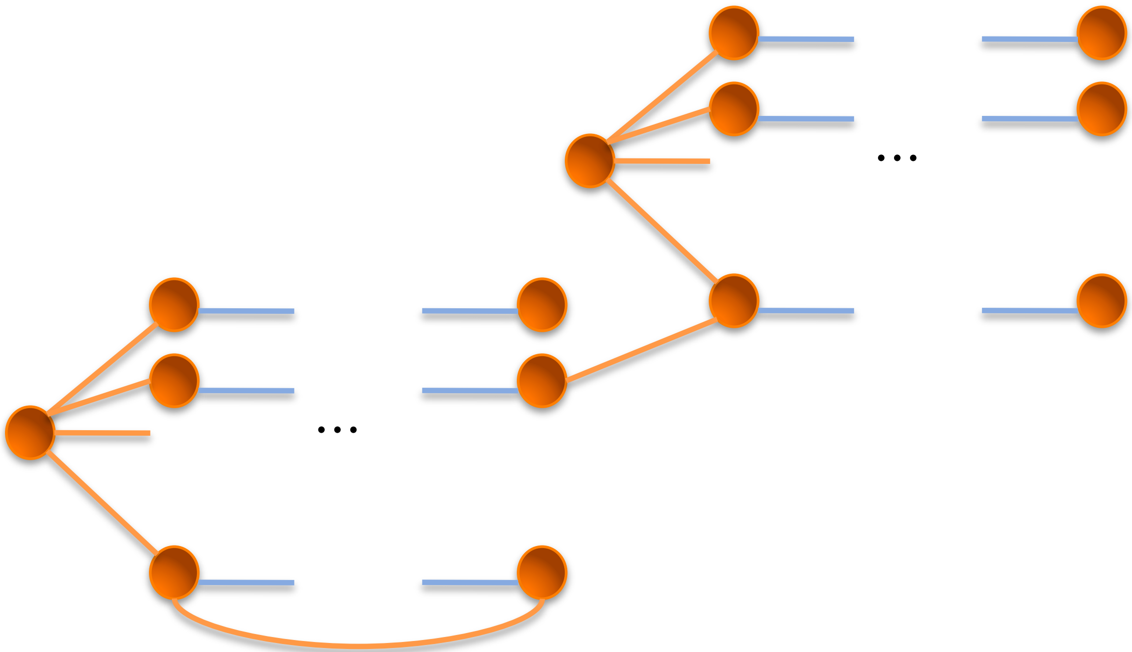
(segue)

Accoppiamento Max in Grafi Qualunque (8)

Algoritmo di Edmonds – segue

4. tutti i nodi esterni sono adiacenti a soli nodi interni:

   M è massimo.



Accoppiamento Max in Grafi Qualunque (9)

Lemma. Ad ogni passo dell'algoritmo di Edmonds, o cresce la dimensione di F, o decresce la dimensione di G, o si trova un cammino aumentante, o M è massimo.

Complessità. Num. di terazioni $\leq$

num. delle crescite di F (al più n)+

num. delle contrazioni di boccioli (al più n)+

num. dei cammini aumentanti (al più $n/2$).

La complessità dipende dalla gestione dei boccioli. A seconda delle versioni: $O(n^3)$ o $O(mn^2)$.

Migliore complessità: $O(m\sqrt{n})$ [Micali & Vazirani '80]