Aplicaţie

Flux maxim $\rightarrow$ cuplaj maxim

în grafuri bipartite

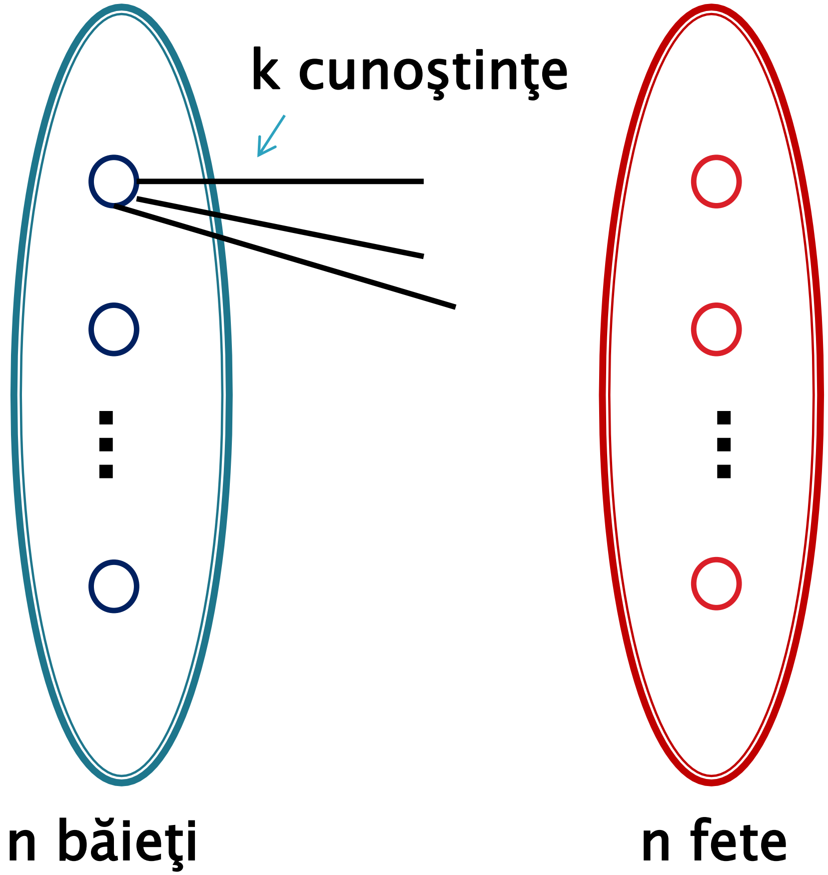
Cuplaje

Repere istorice. Aplicaţii

- **Problema seratei (perechilor)** – sec XIX
  - n băieţi, n fete
  - Un băiat cunoaşte exact k fete
  - O fată cunoaşte exact k băieţi

Repere istorice. Aplicaţii

- Problema seratei (perechilor)



Repere istorice. Aplicaţii

- **Problema seratei (perechilor)** – sec XIX
  - Se poate organiza o repriză de dans astfel încât fiecare participant să danseze cu o cunoştinţă a sa?

Repere istorice. Aplicaţii

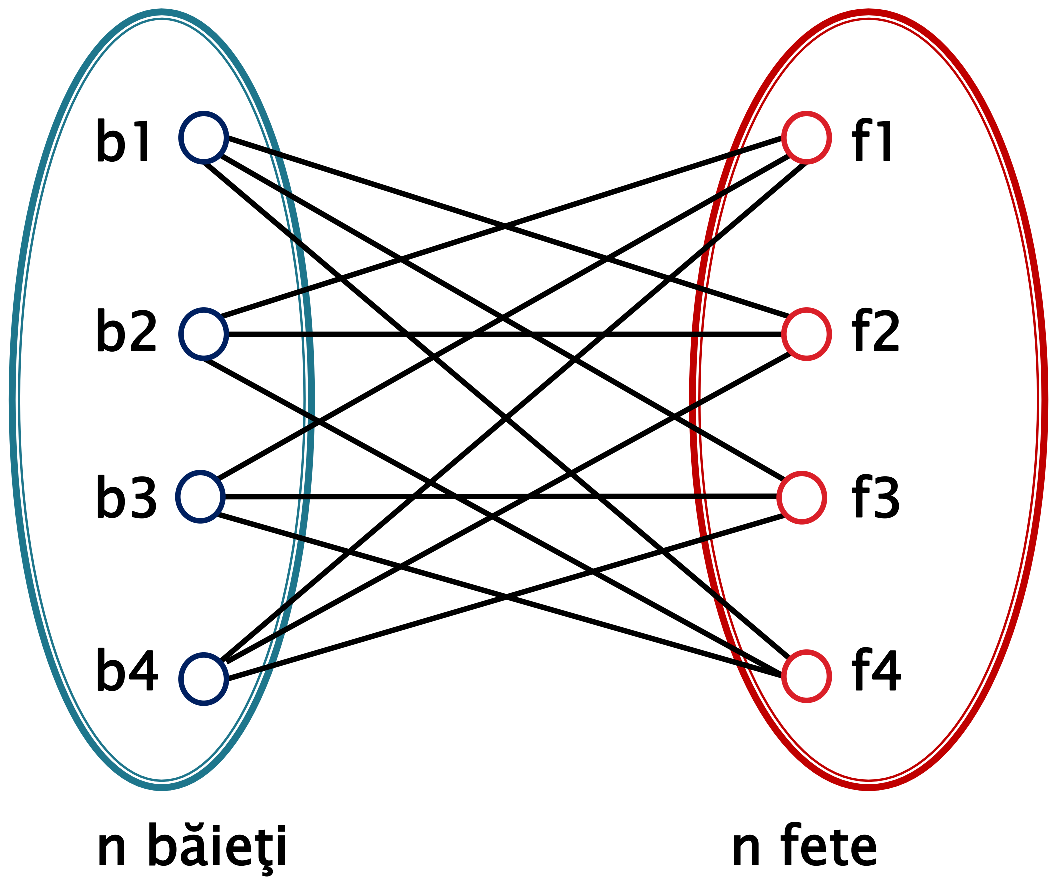
- **Problema seratei (perechilor)** – sec XIX
  - Se poate organiza o repriză de dans astfel încât fiecare participant să danseze cu o cunoştinţă a sa?
  - Se pot organiza k reprize de dans în care fiecare participant să danseze câte un dans cu fiecare cunoştinţă a sa?

Repere istorice. Aplicaţii

- **Problema seratei (perechilor)** – sec XIX

n=4

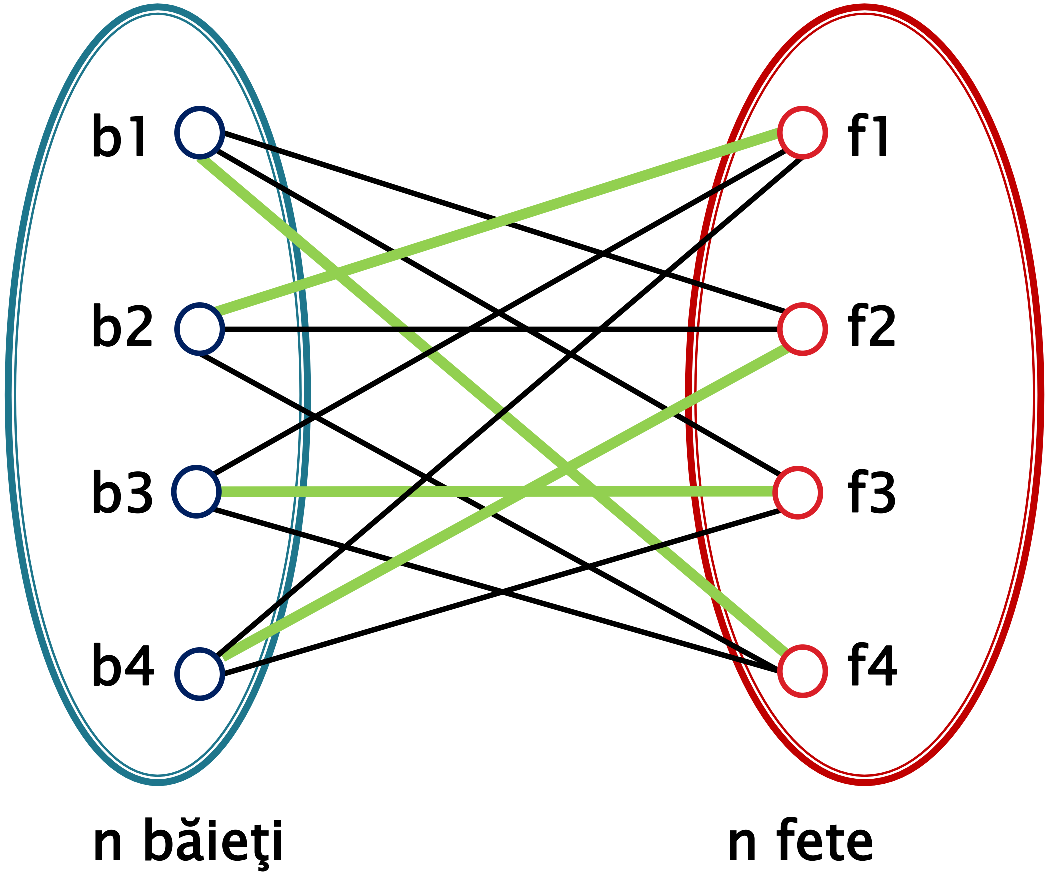
k=3



Repere istorice. Aplicaţii

- O repriză de dans

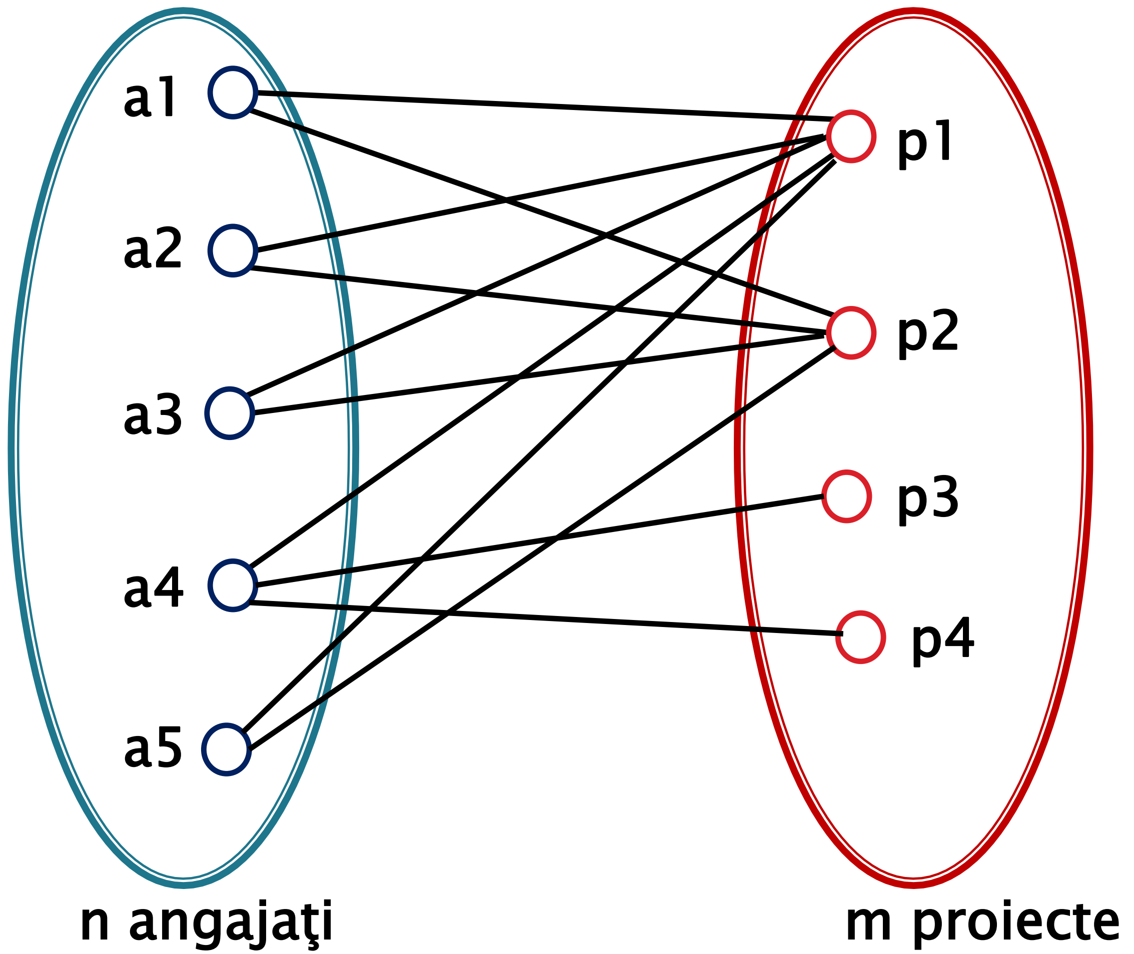
b1,f4
b2,f1
b3,f3
b4,f2



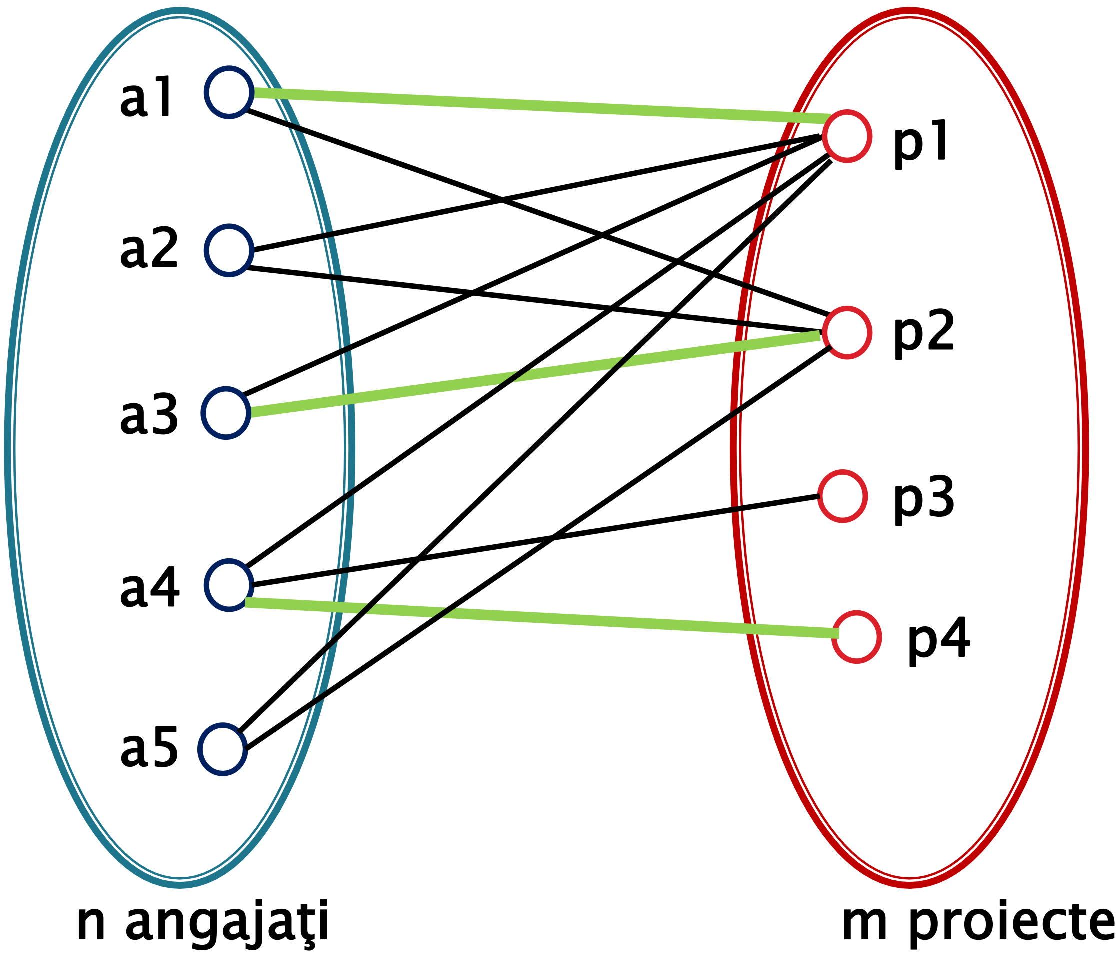
Repere istorice. Aplicaţii

- **Organizare de competiții**
- **Probleme de repartiţie**
  - lucrători – locuri de muncă
  - profesori – examene /conferinţe
    - **Problema orarului**

Alte aplicaţii

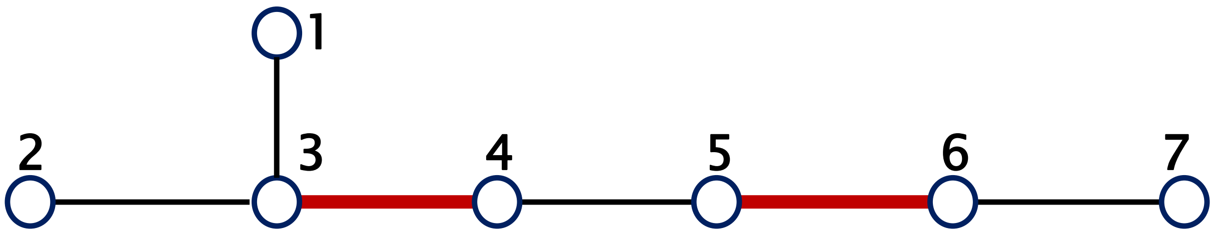


Alte aplicaţii



Fie $G = (V, E)$ un graf şi $\mathbf{M} \subseteq \mathbf{E}$.

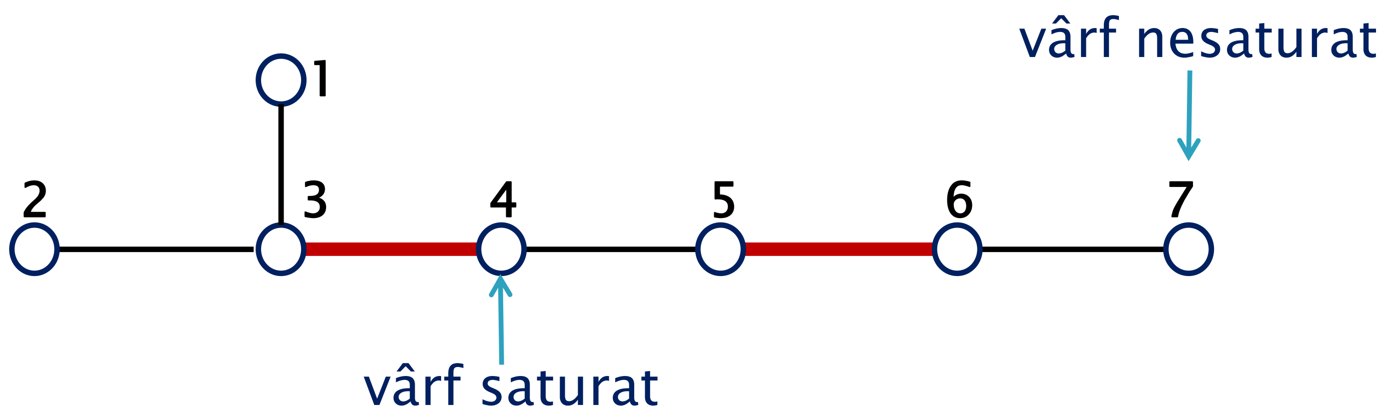
- M s.n **cuplaj** dacă orice două muchii din M sunt neadiacente



cuplaj M={ {3,4}, {5,6} }

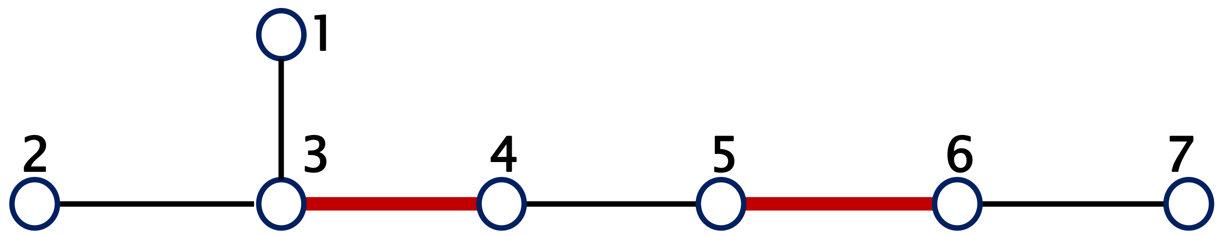
Fie G = (V, E) un graf şi **M** $\subseteq$ **E**.

- M s.n **cuplaj** dacă orice două muchii din M sunt neadiacente
- V(M) = mulţimea vârfurilor **M-saturate**
- V(G) – V(M) = mulţimea vârfurilor **M-nesaturate**

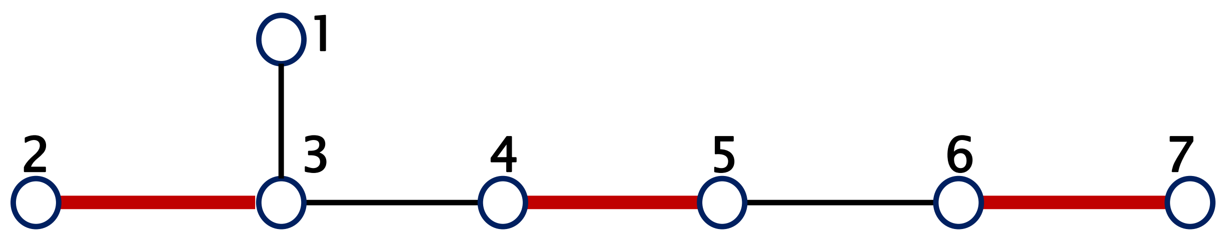


- Un cuplaj M^* s.n **cuplaj de cardinal maxim** (**cuplaj maxim**):

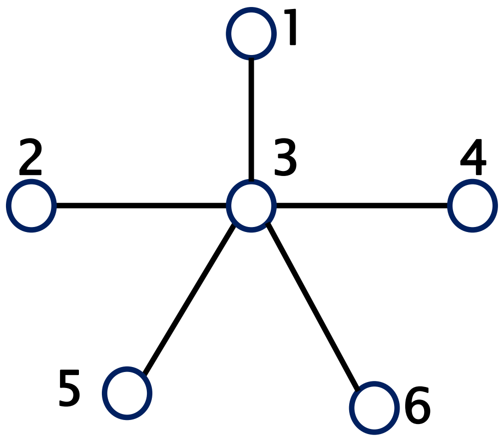
$$| M^*| \geq |M|, \forall M \subseteq E \text{ cuplaj}$$



cuplaj



cuplaj de cardinal maxim



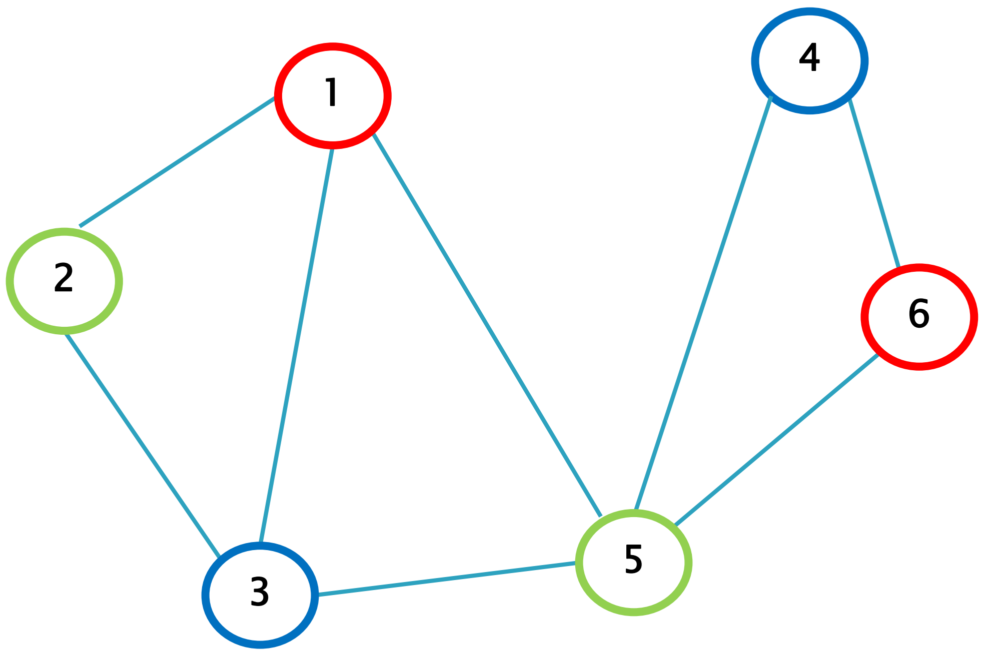
cuplaj de cardinal maxim ?

Grafuri bipartite

Colorări ale grafurilor

- $G = (V, E)$ graf neorientat
  - $c : V \rightarrow \{1, 2, \ldots, p\}$ s.n <u>p-colorare</u> a lui G
  - $c : V \rightarrow \{1, 2, \ldots, p\}$ cu $c(x) \neq c(y)\ \forall xy \in E$ s.n **<u>p-colorare proprie</u>** a lui G
  - G s.n <u>p-colorabil</u> dacă admite o p-colorare proprie

Colorări ale grafurilor



3-colorabil, dar nu și 2-colorabil (!)

Graf bipartit

- $G = (V, E)$ graf neorientat s.n. **bipartit** $\Leftrightarrow$ există o partiție a lui V în două submulțimi V_1, V_2 (**bipartiție**):

$$V = V_1 \cup V_2$$

$$V_1 \cap V_2 = \emptyset$$

astfel încât orice muchie $e \in E$ are o extremitate în V_1 și cealaltă în V_2

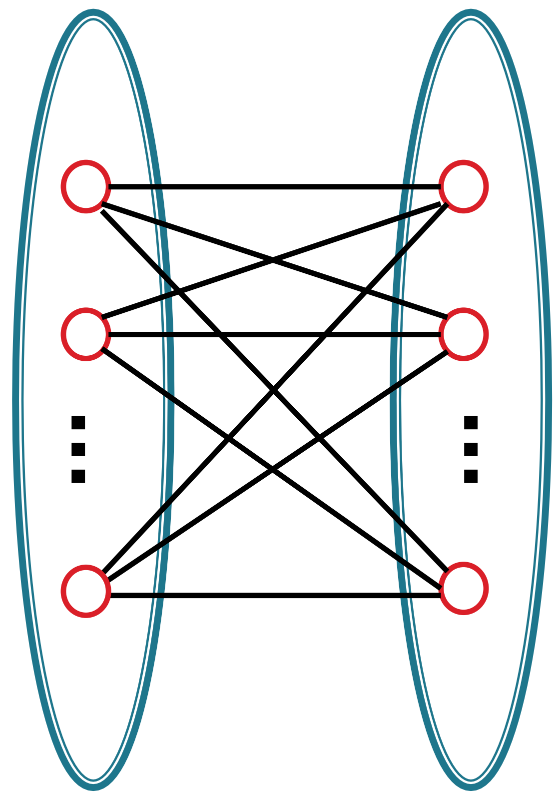
- Notăm $G = (V_1 \dot{\cup} V_2, E)$

Graf bipartit

- G = (V, E) s.n **bipartit complet** $\Leftrightarrow$

  este bipartit și $E = \{xy \mid x \in V_1, y \in V_2\}$

- Notăm cu $K_{p,q}$ dacă $p = |V_1|$ și $q = |V_2|$



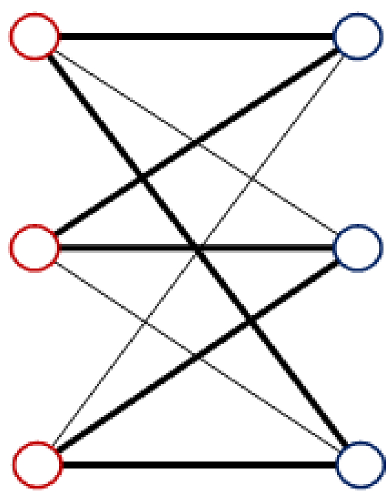
Graf bipartit

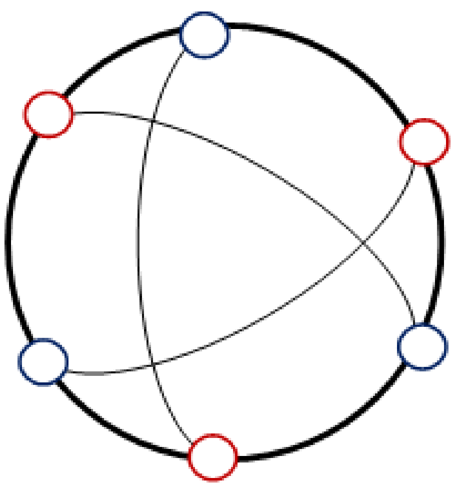
- G = (V, E) s.n **bipartit complet** $\Leftrightarrow$

  este bipartit și $E = \{xy \mid x \in V_1, y \in V_2\}$

- Notăm cu $K_{p,q}$ dacă $p = |V_1|$ și $q = |V_2|$

- $K_{3,3}$





Graf bipartit

Observație

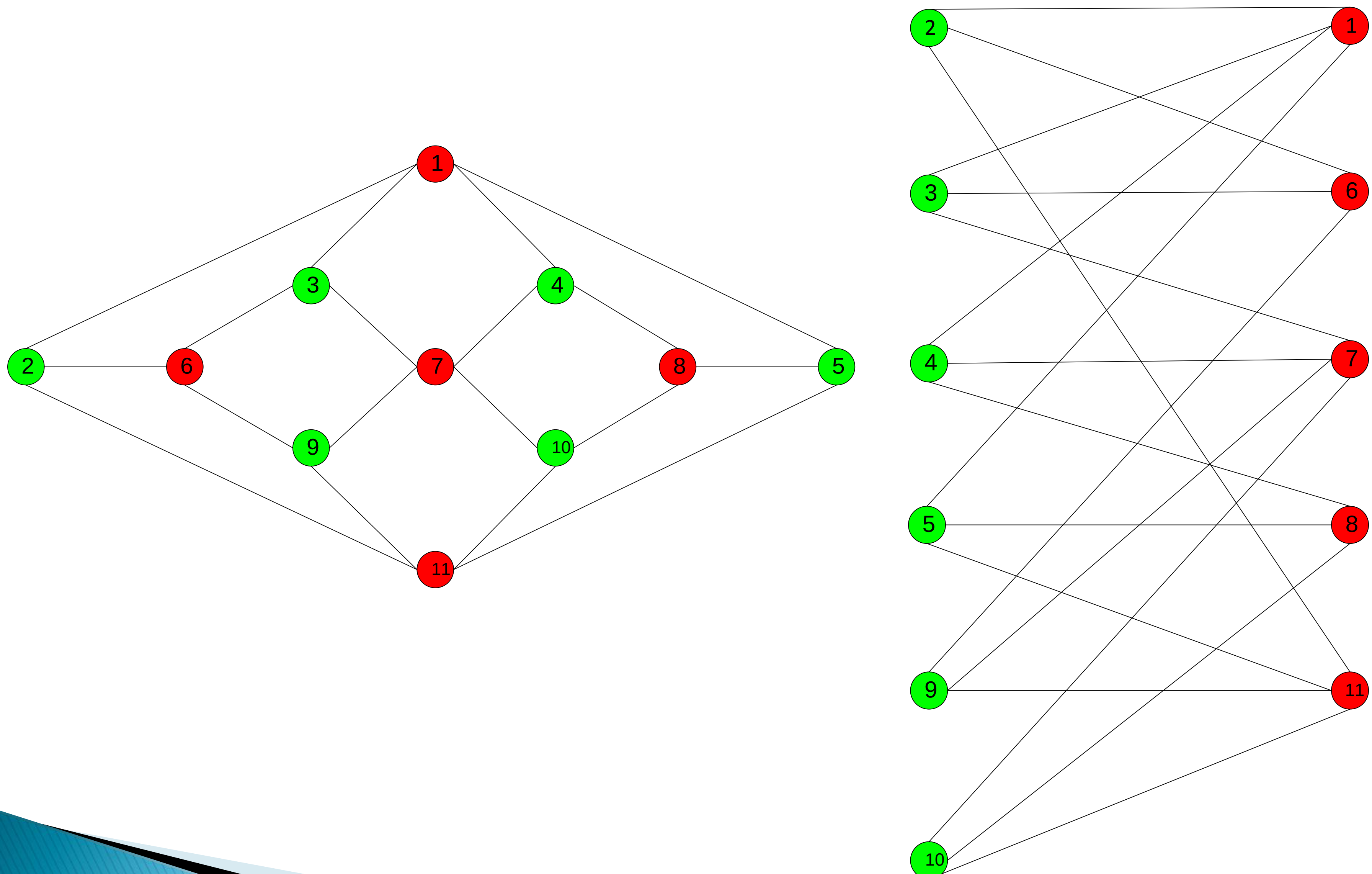
- $G = (V, E)$ **bipartit** $\Leftrightarrow$

  există o 2-colorare proprie a vârfurilor (**bicolorare**):

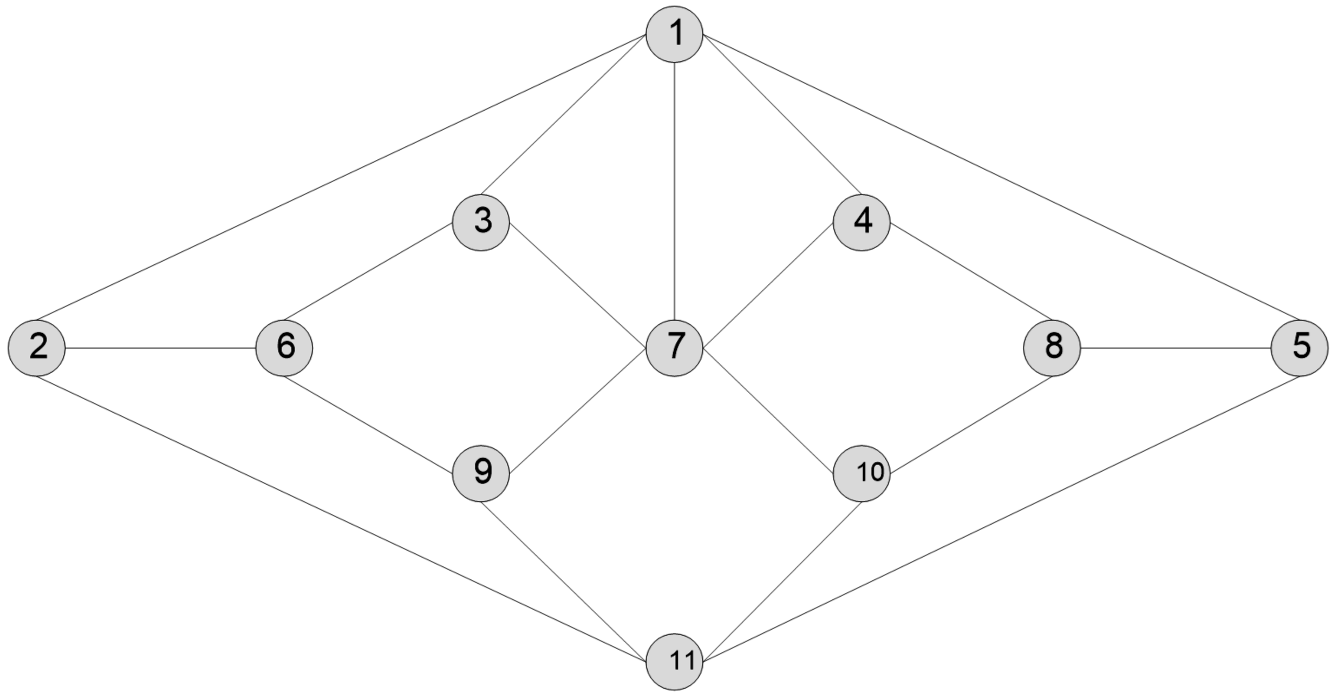
  $$c : V \rightarrow \{1, 2\}$$

  (i.e. astfel încât pentru orice muchie $e=xy\in E$ avem $c(x) \neq c(y)$)

Graf bipartit

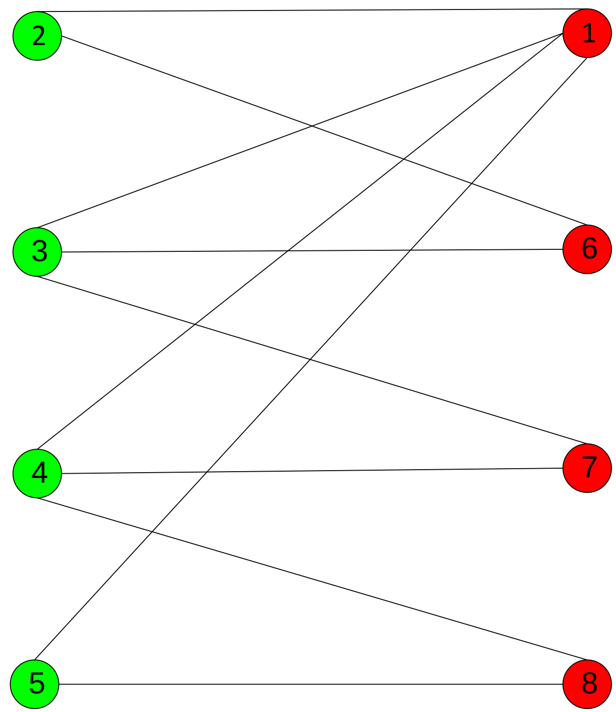


Graf bipartit



nu este bipartit

Modelare

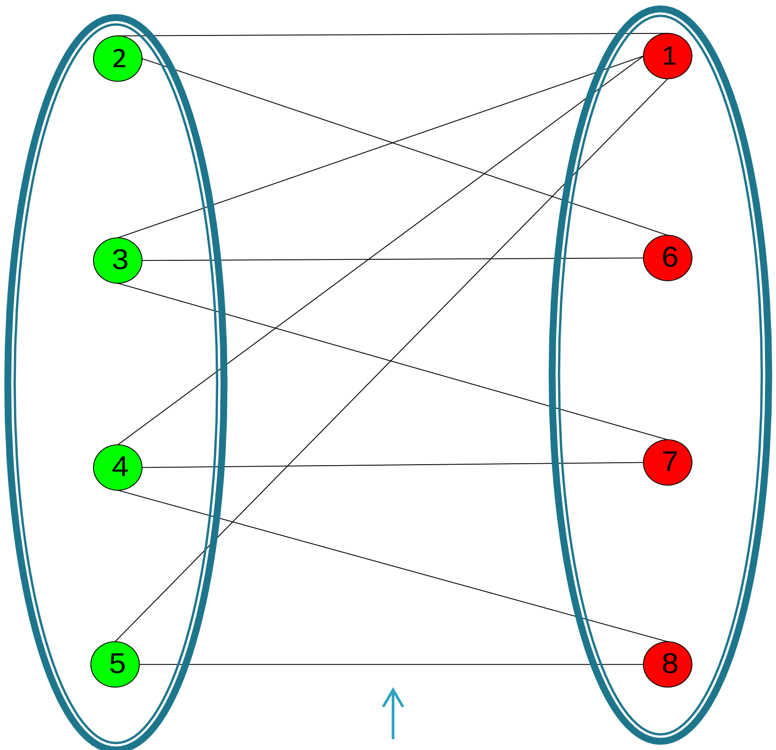


Profesori *predau la* **Cursuri**

Candidați *depun CV la* **Joburi**

Aplicații

Graf de conflicte (exemplu substanțe care interacționează, activități incompatibile, relații în rețele sociale)



submulțime fără conflicte
(de evenimente compatibile)

submulțime fără conflicte
(de evenimente compatibile)

în conflict cu

- Cuplaje, rețele…

Aplicații p -colorări

Exemplu – De câte săli este nevoie minim pentru programarea într-o zi a n conferințe cu intervale de desfășurare date?

Conf. 1: interval (1,4)

Conf. 2: interval (2,3)

Conf. 3: interval (2,5)

Conf. 4: interval (6,8)

Conf. 5: interval (3,8)

Conf. 6: interval (6,7)

Aplicații p -colorări

Exemplu – De câte săli este nevoie minim pentru programarea într-o zi a n conferințe cu intervale de desfășurare date?

Conf. 1: interval (1,4)

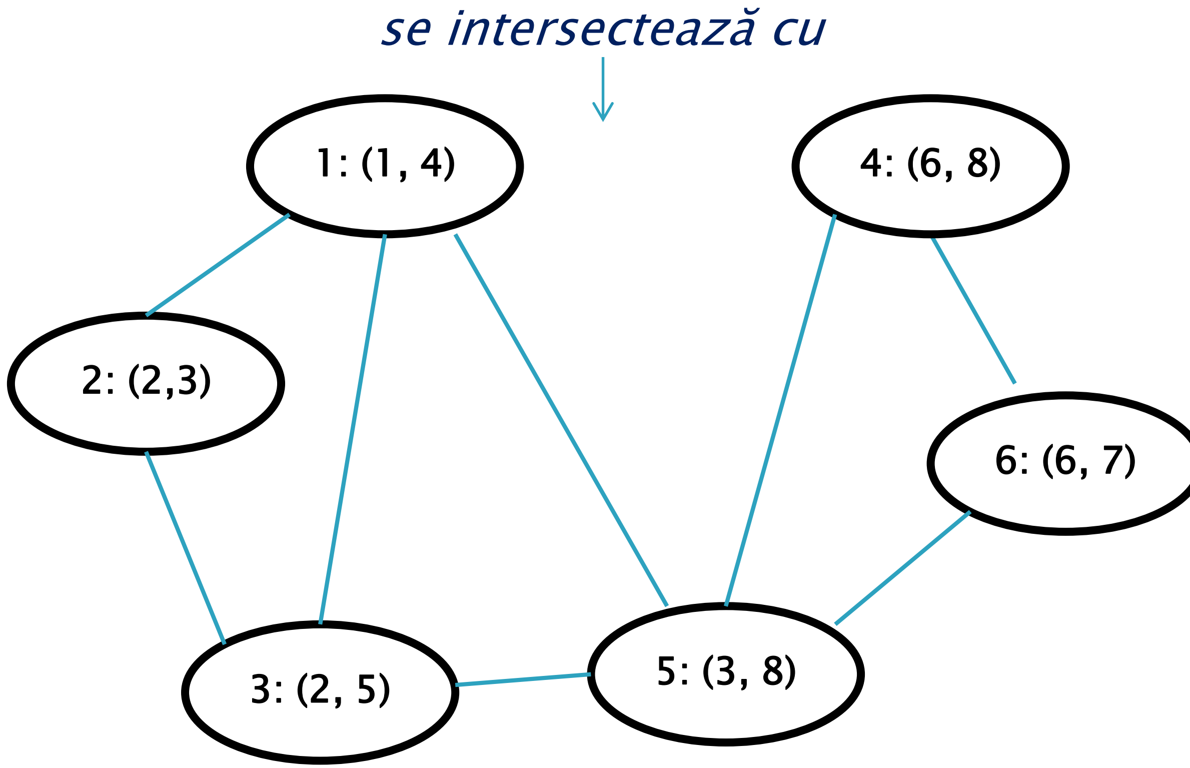
Conf. 2: interval (2,3)

Conf. 3: interval (2,5)

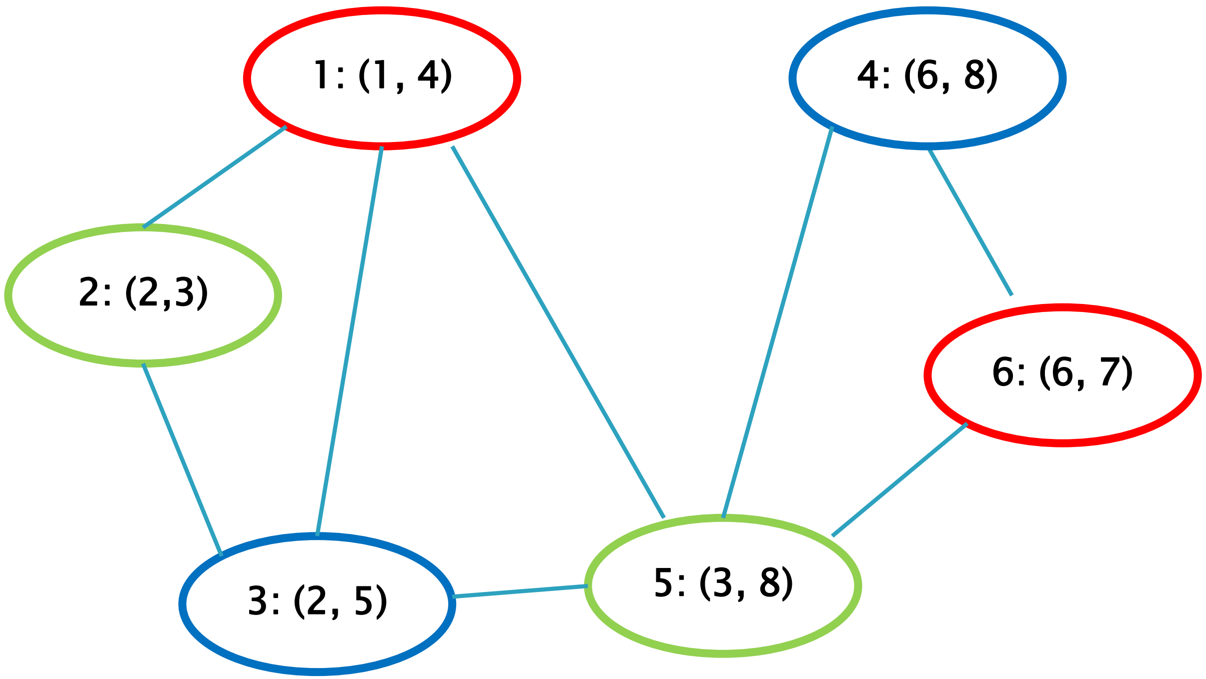
Conf. 4: interval (6,8)

Conf. 5: interval (3,8)

Conf. 6: interval (6,7)



Graful intersecției intervalelor este 3-colorabil:



Sunt necesare minim 3 săli (corespunzătoare celor 3 culori):

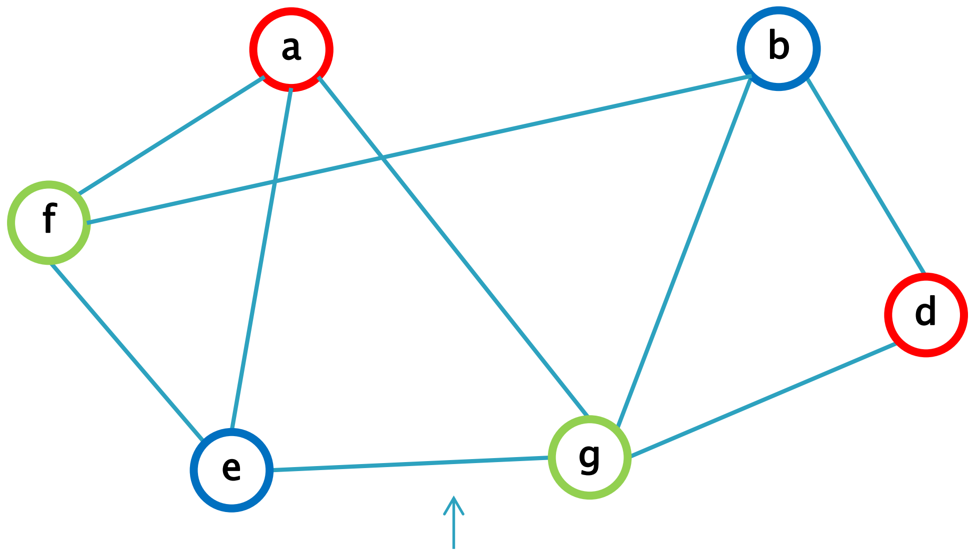
Sala 1: (1,4), (6,7)

Sala 2: (2,3), (3,8)

Sala 3: (2,5), (6,8)

Aplicații p -colorări

Alocare de regiştrii (Register allocation problem)



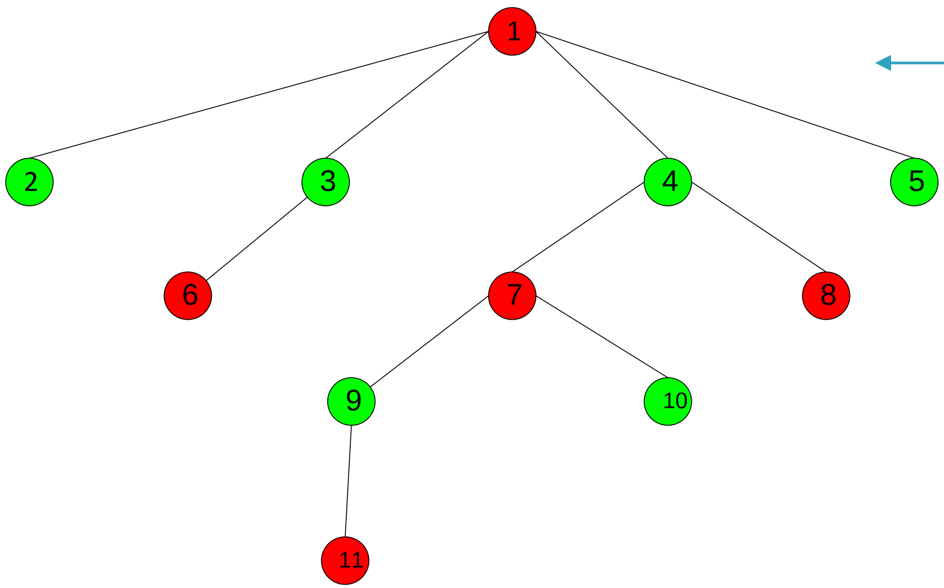
pot fi simultan active
(nu pot fi memorate în acelaşi registru)

- Numărul de culori = numărul de regiştri
- Vârfuri de aceeaşi culoare = pot fi memorate în acelaşi registru

Graf bipartit

- **Propoziție**

Un arbore este graf bipartit



– Fixăm o rădăcină

– Colorăm alternativ nivelurile

Graf bipartit

- **Teorema König – Caracterizarea grafurilor bipartite**

Fie $G = (V, E)$ un graf cu $n \geq 2$ vârfuri.

Avem

G este bipartit $\Leftrightarrow$ toate ciclurile elementare din G sunt pare

Graf bipartit

- **Teorema König – Caracterizarea grafurilor bipartite**

Fie $G = (V, E)$ un graf cu $n \geq 2$ vârfuri.

Avem

$$G \text{ este bipartit} \Leftrightarrow \text{toate ciclurile elementare din } G \text{ sunt pare}$$

Graf bipartit

- **Teorema König – Caracterizarea grafurilor bipartite**

Demonstrație $\Rightarrow$ Evident, deoarece un ciclu impar nu poate fi colorat propriu cu două culori.

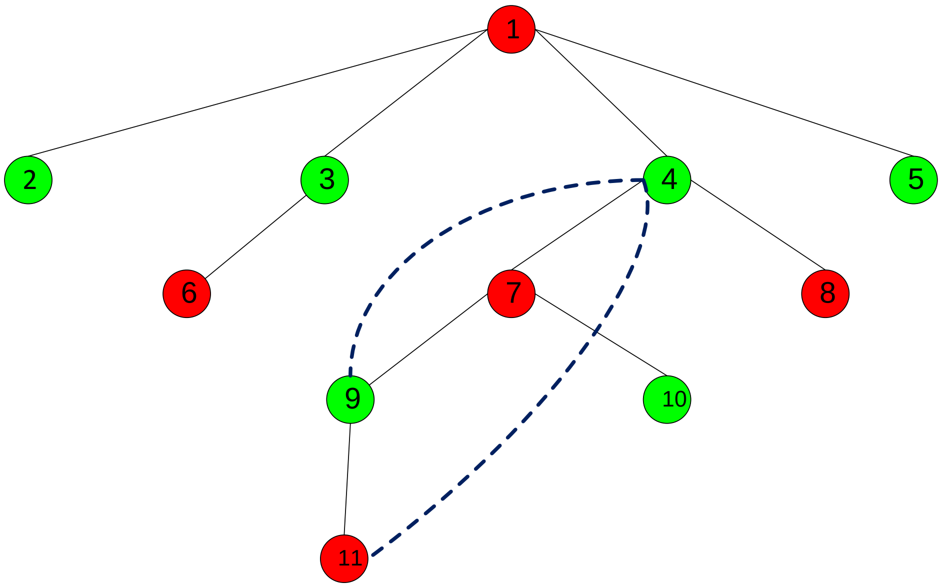
Graf bipartit

- **Teorema König – Caracterizarea grafurilor bipartite**

Demonstrație ⇐ Presupunem G conex.

Colorăm propriu cu 2 culori un arbore parțial T al său.

Orice altă muchie uv din graf are extremitățile colorate diferit deoarece



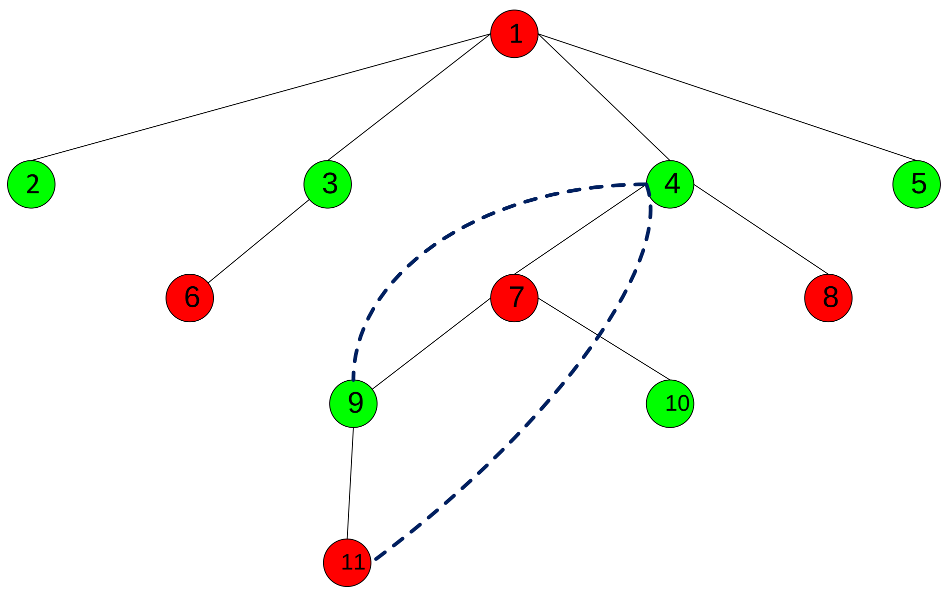
Graf bipartit

- **Teorema König – Caracterizarea grafurilor bipartite**

Demonstrație $\Leftarrow$ Presupunem G conex.

Colorăm propriu cu 2 culori un arbore parțial T al său.

Orice altă muchie uv din graf are extremitățile colorate diferit deoarece formează un ciclu elementar cu lanțul de la u la v din arbore și acest ciclul are lungime pară, deci u și v se află pe niveluri de paritate diferită în T

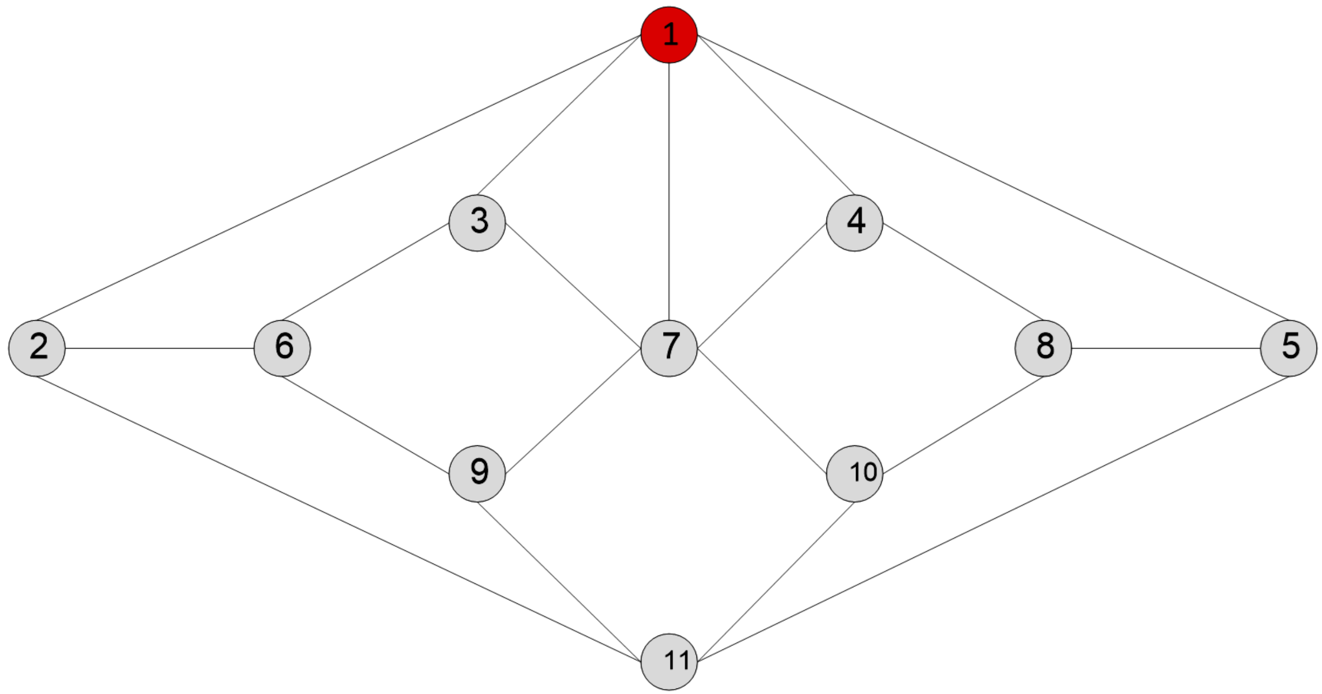


Bibliografie DR Popescu – Combinatorică și Teoria grafurilor (Teorema 4.18)

Graf bipartit

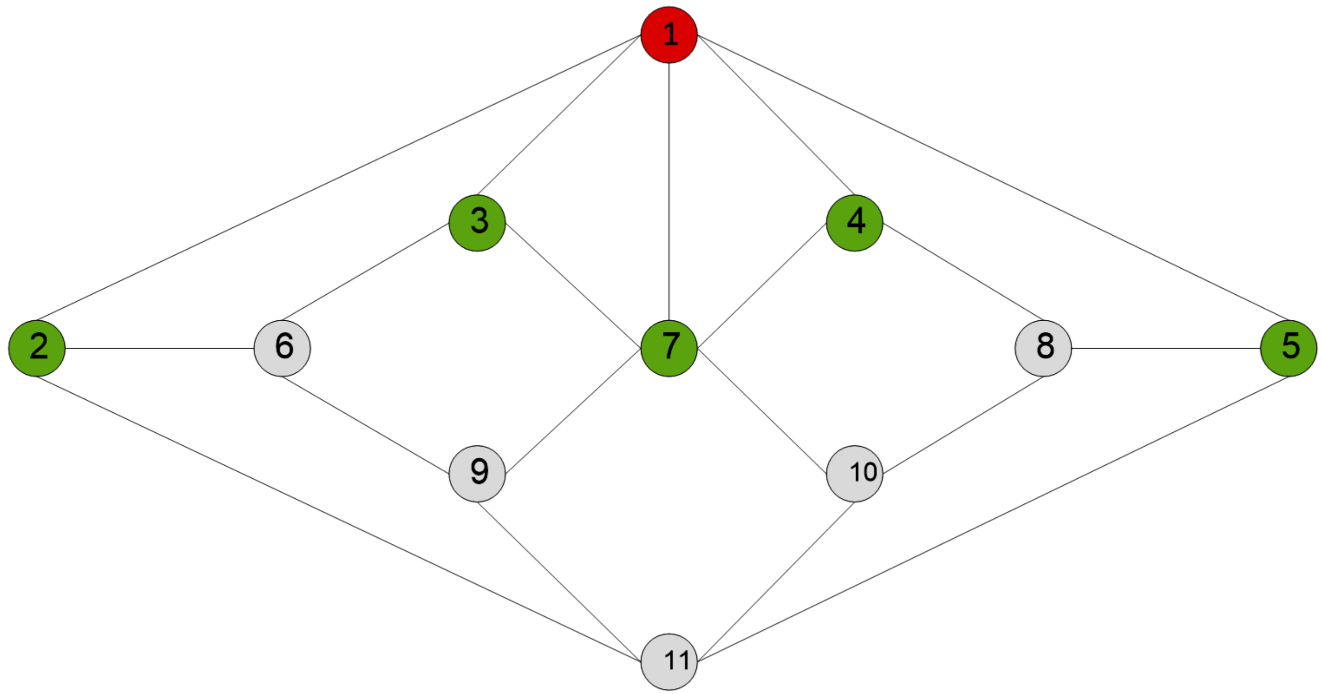
- **Teorema König ⇒ Agoritm pentru a testa dacă un graf este bipartit**
  - Colorăm (propriu) cu 2 culori un arbore parțial al său printr-o **parcurgere** (colorăm orice vecin j nevizitat al vârfului curent i cu o culoare diferită de cea a lui i)
  - Testăm dacă celelalte muchii - de la i la **vecini j deja vizitați** (colorați) au extremitățile i și j colorate diferit

Exemplu test bipartit BF

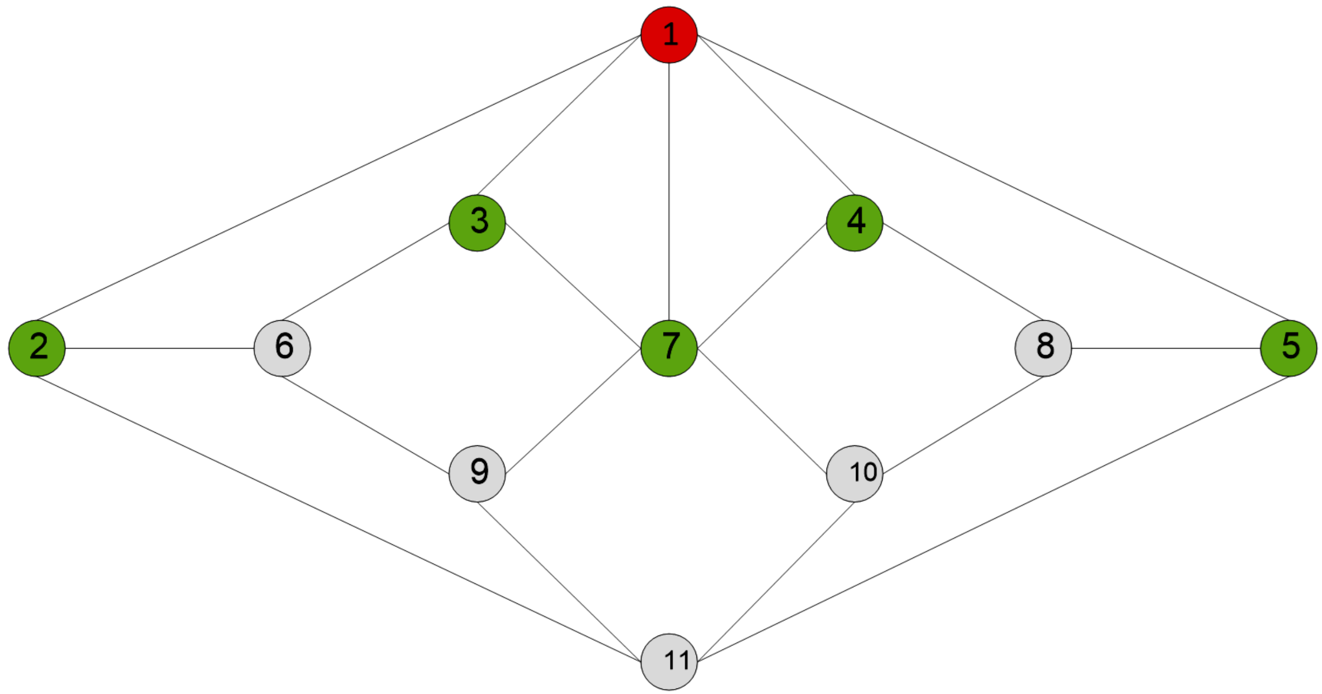


$i = 1$

Exemplu test bipartit BF

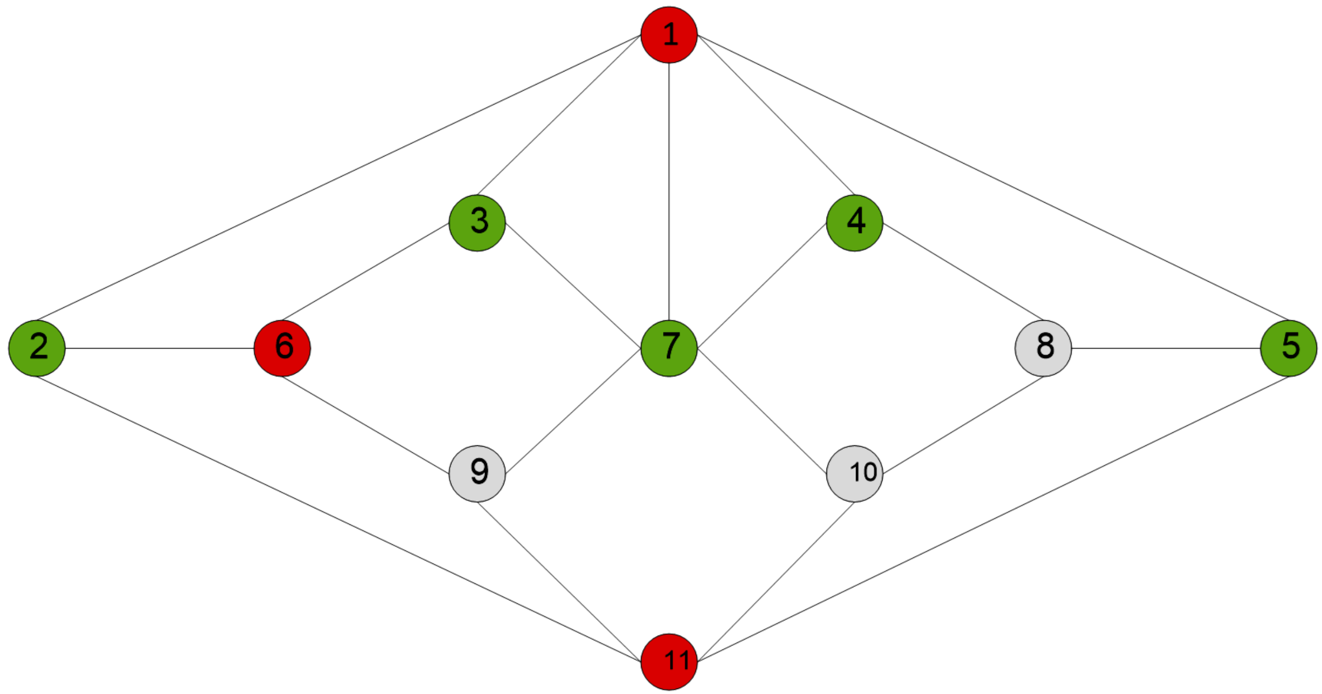


Exemplu test bipartit BF

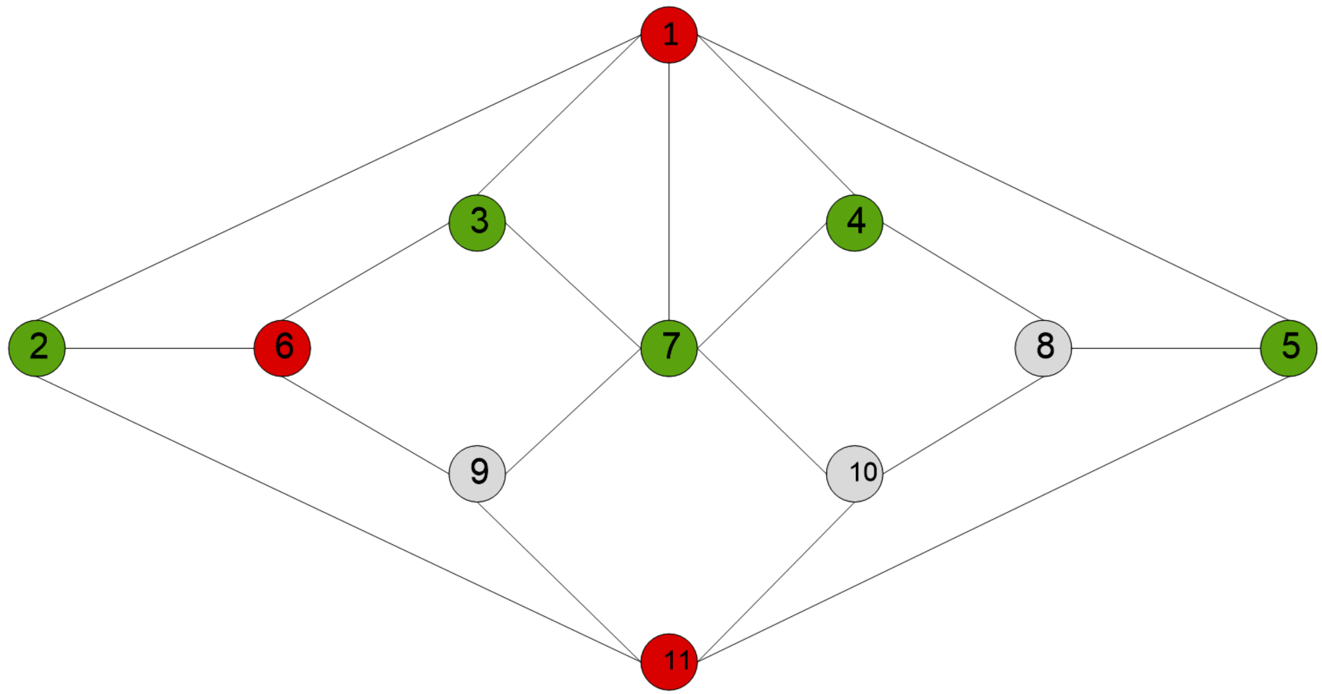


i = 2

Exemplu test bipartit BF

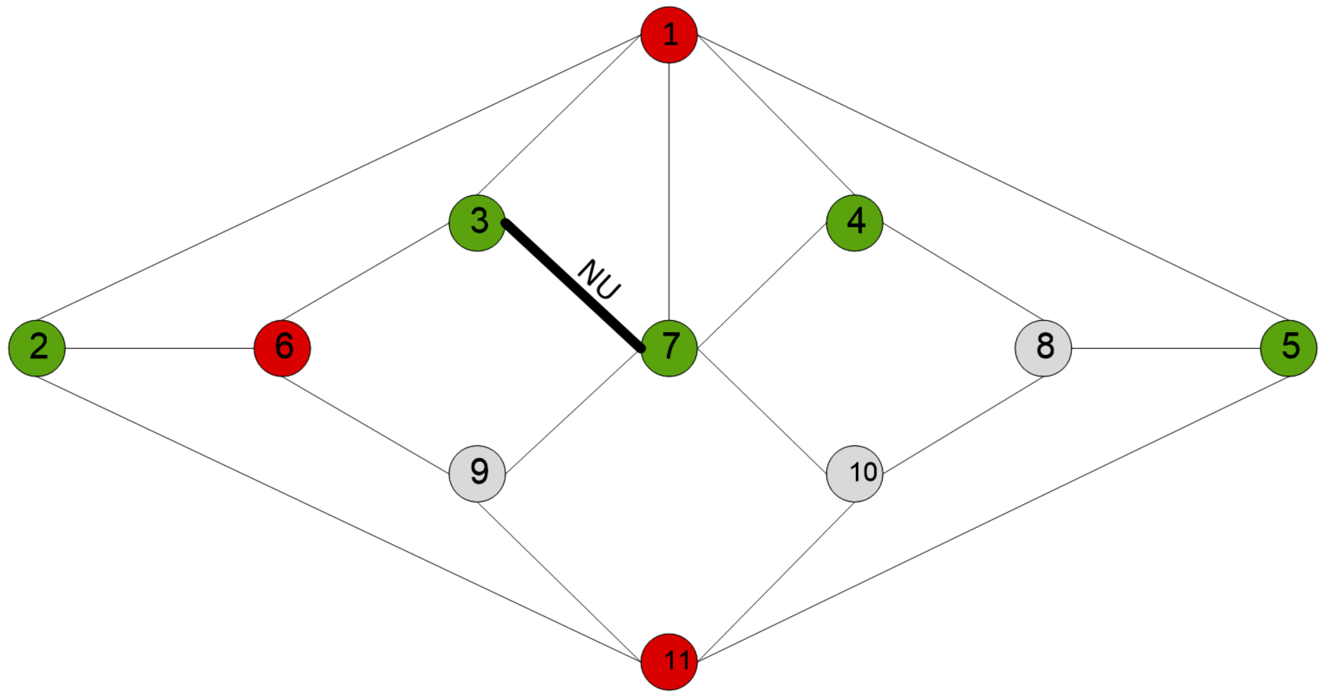


Exemplu test bipartit BF



$i = 3$

Exemplu test bipartit BF



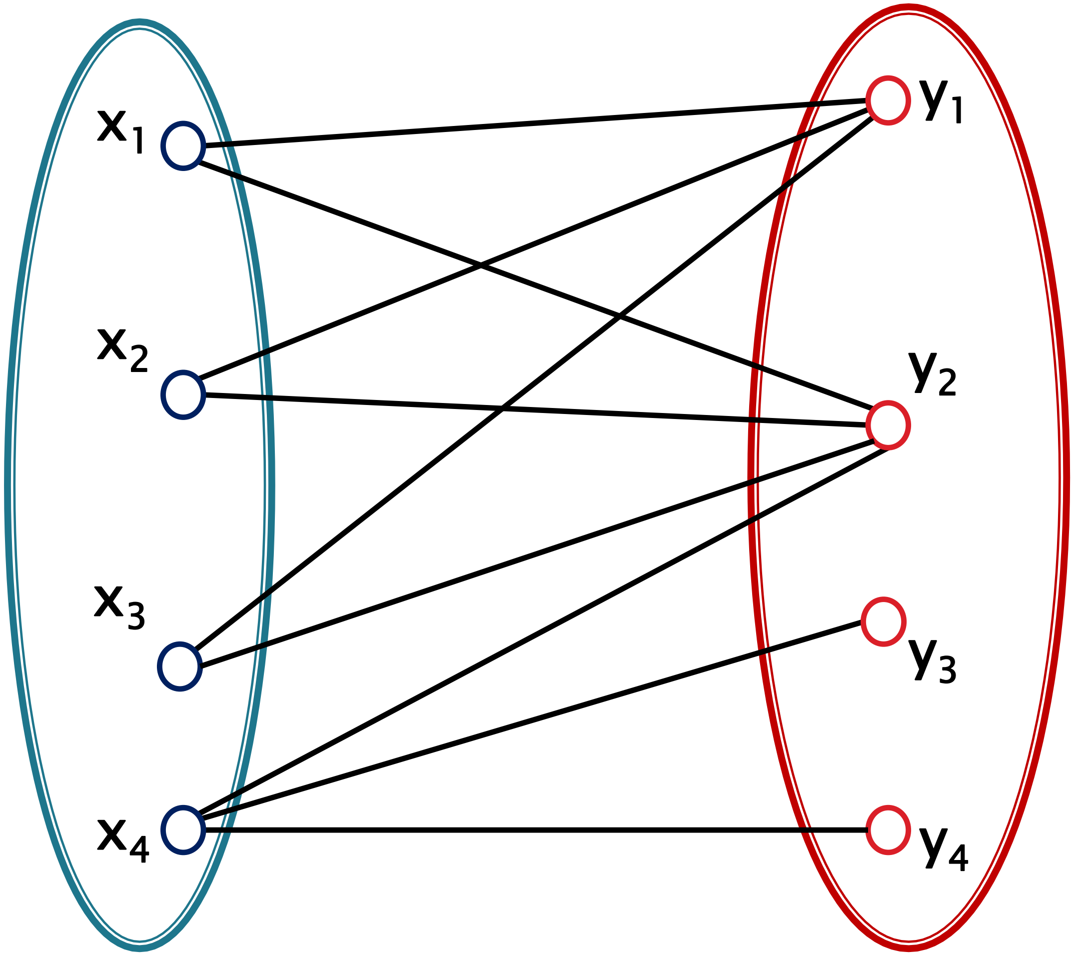
Algoritm

Flux maxim $\rightarrow$ cuplaj maxim

în graf bipartit

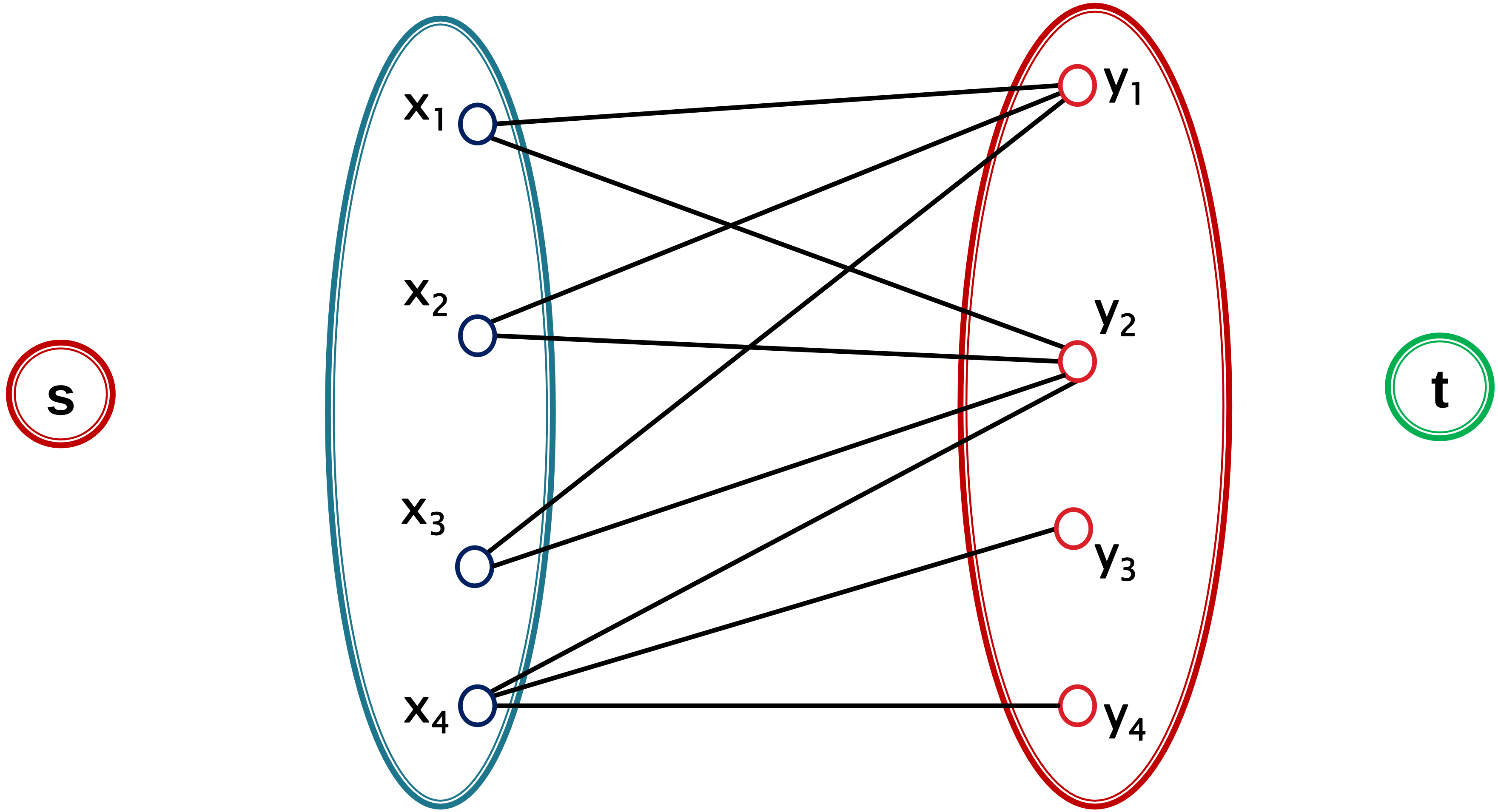
Algoritm de determinare a unui cuplaj maxim într-un graf bipartit

- Reducem problema determinării unui cuplaj maxim într-un cuplaj bipartit G la determinarea unui flux maxim într-o rețea de transport asociată lui G
- Construim rețeaua de transport N_G asociată lui G astfel:

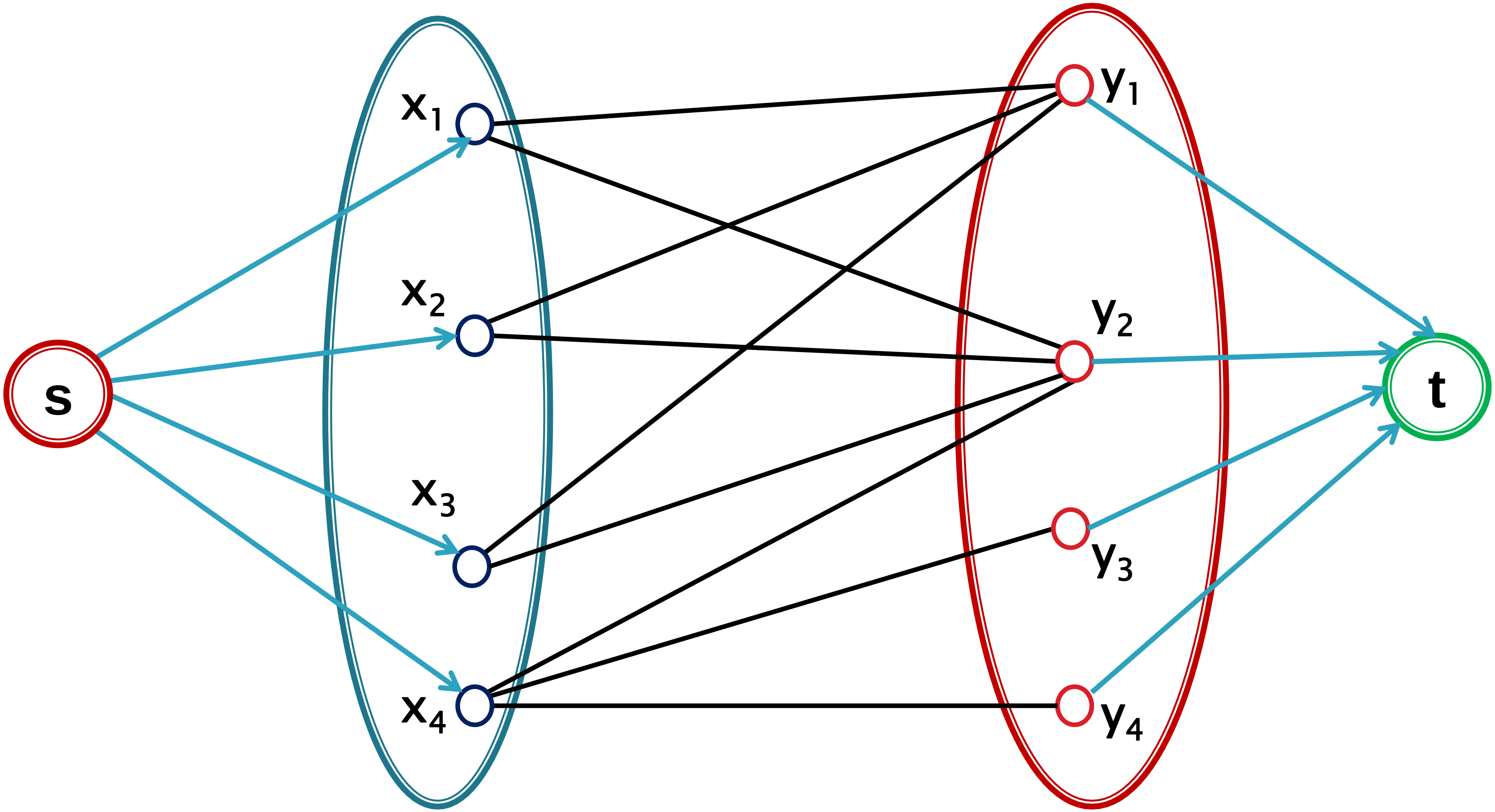


x1
x2
x3
x4
y1
y2
y3
y4

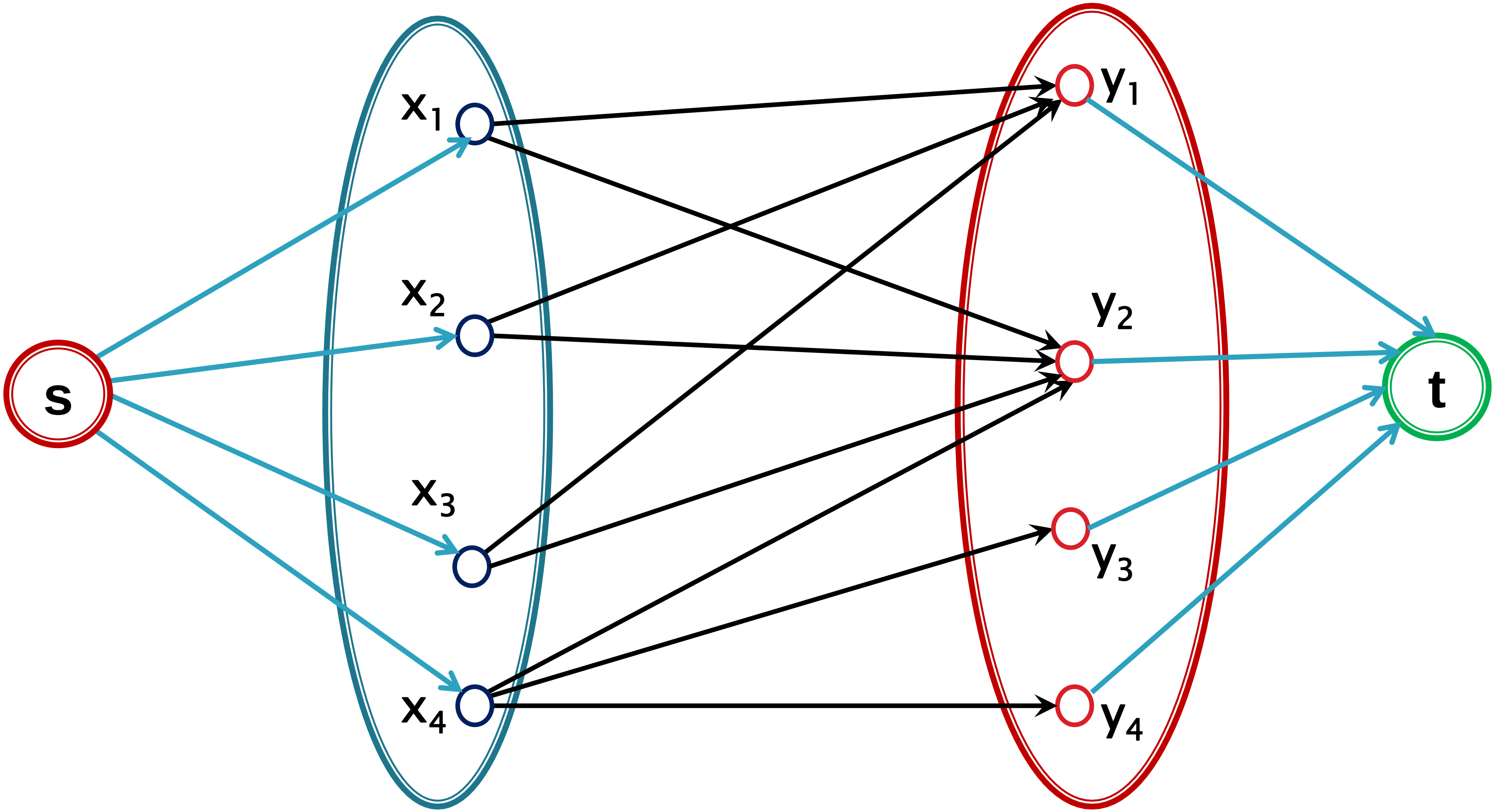
- Adăugăm două noduri noi s şi t



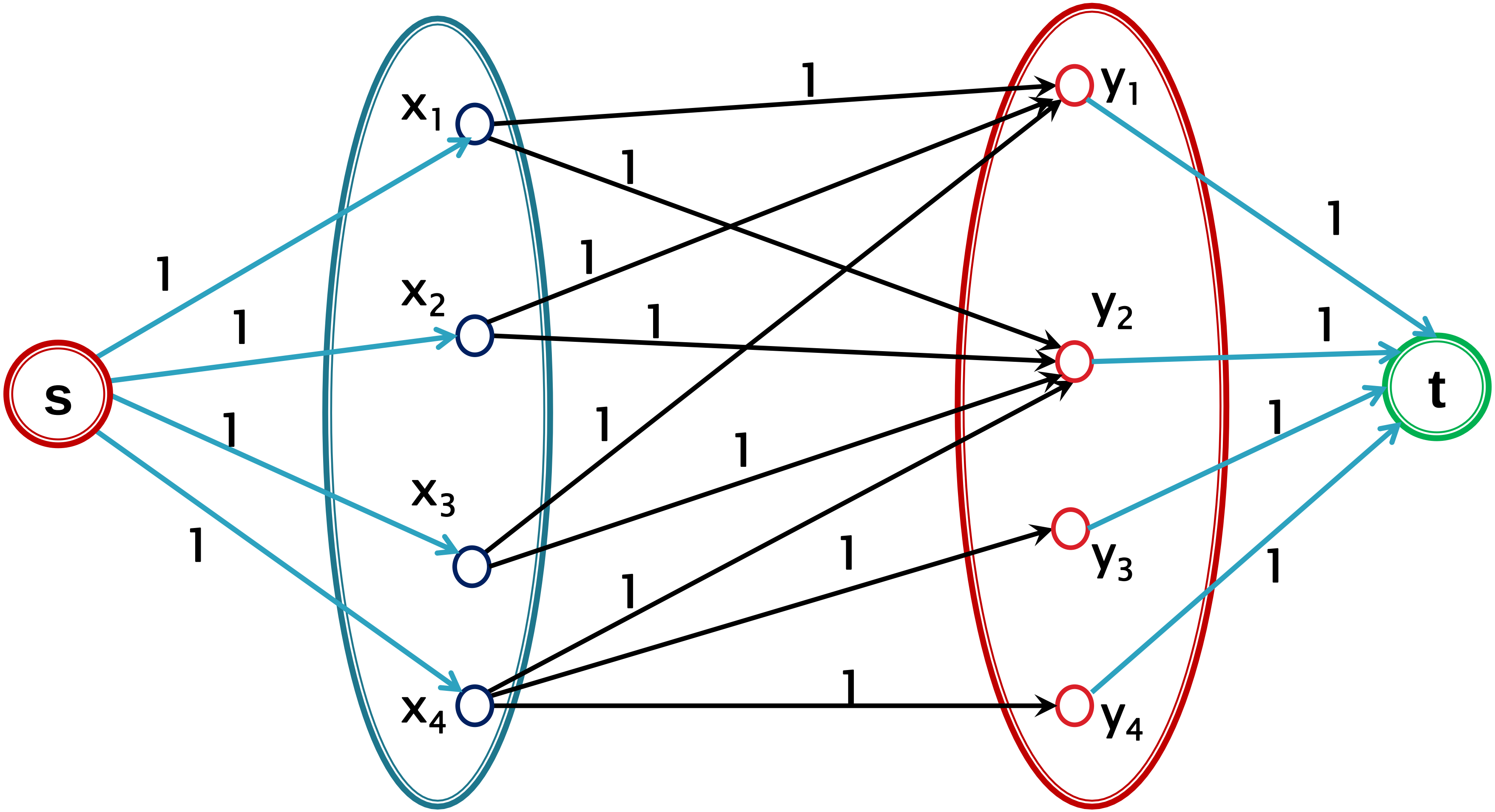
- Adăugăm arce (s,x_i), pentru $x_i \in X$ şi (y_j, t), $y_j \in Y$



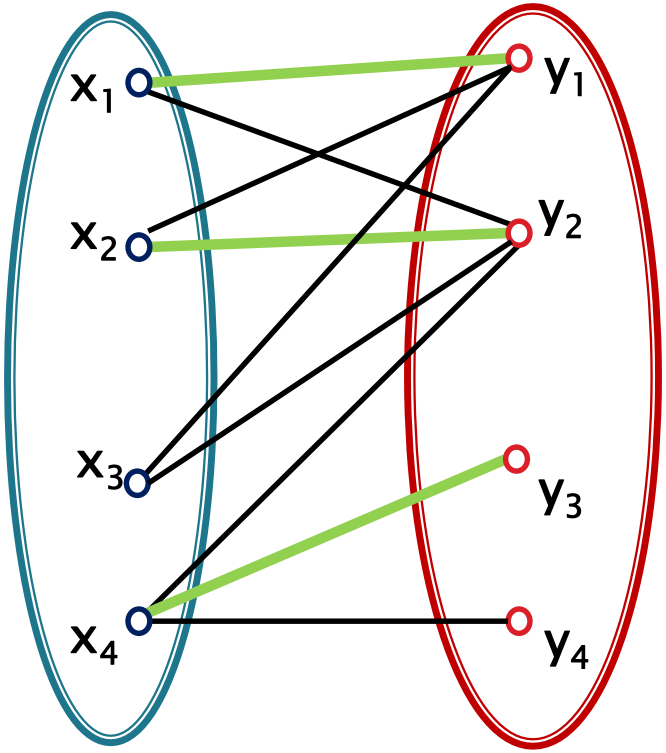
- Transformăm muchiile $x_i y_j$ în arce (de la X la Y)

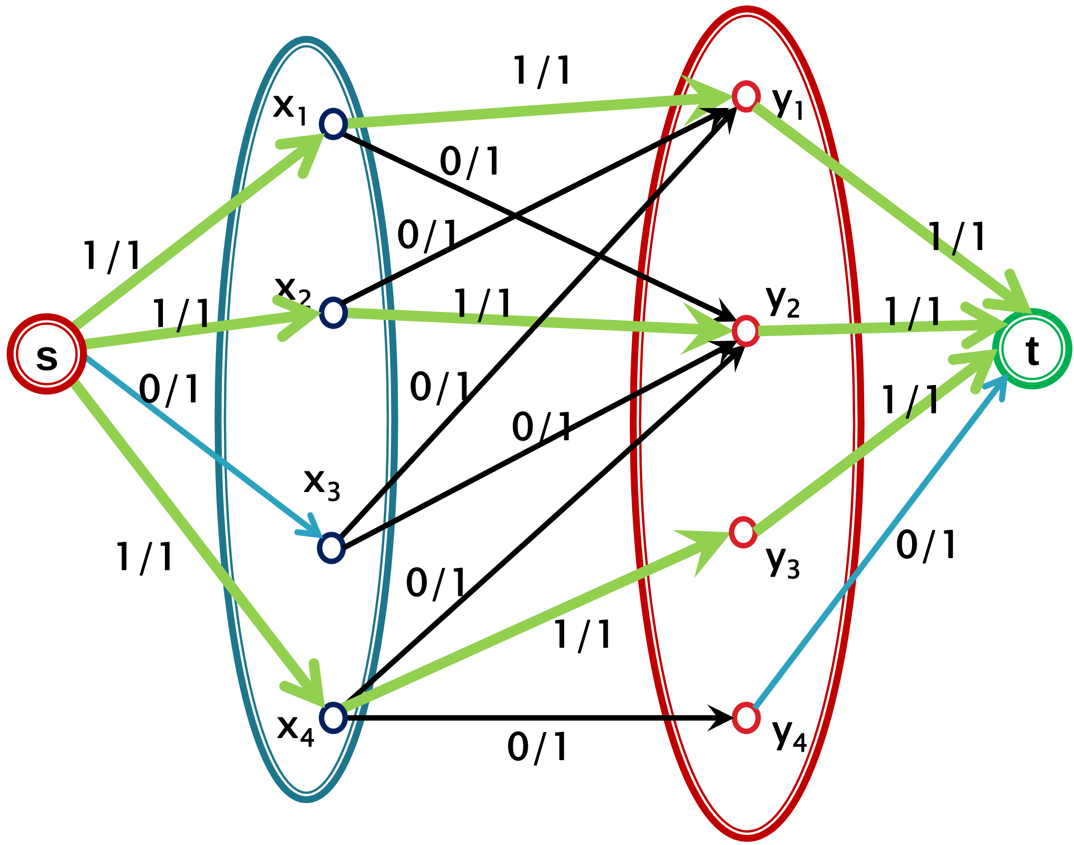


- Asociem fiecărui arc capacitatea 1



- Cuplaj M în G $\Leftrightarrow$ flux f în rețea
- $|M|=val(f)$





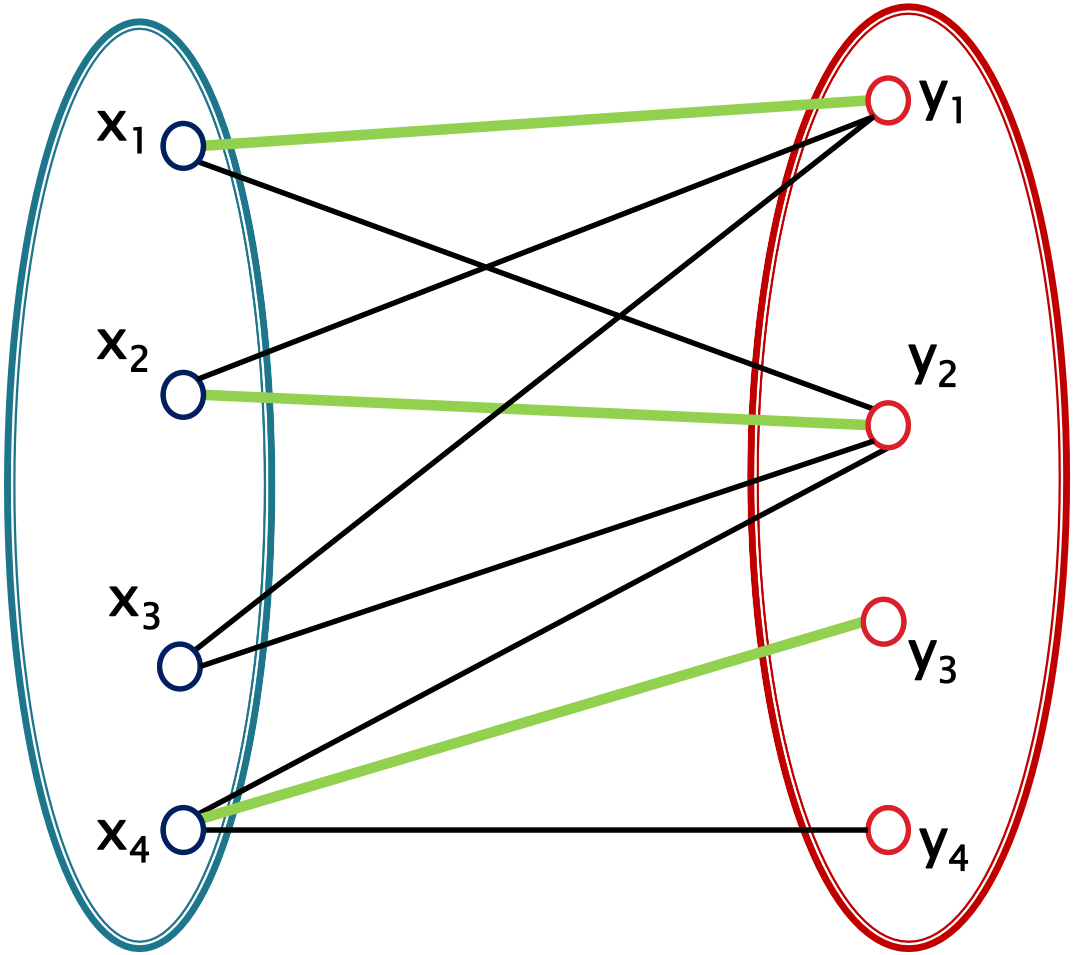
Proprietatea 1

Fie $G=(X \cup Y, E)$ un graf bipartit şi M un cuplaj în G. Atunci există un flux f în reţeaua de transport asociată N_G cu

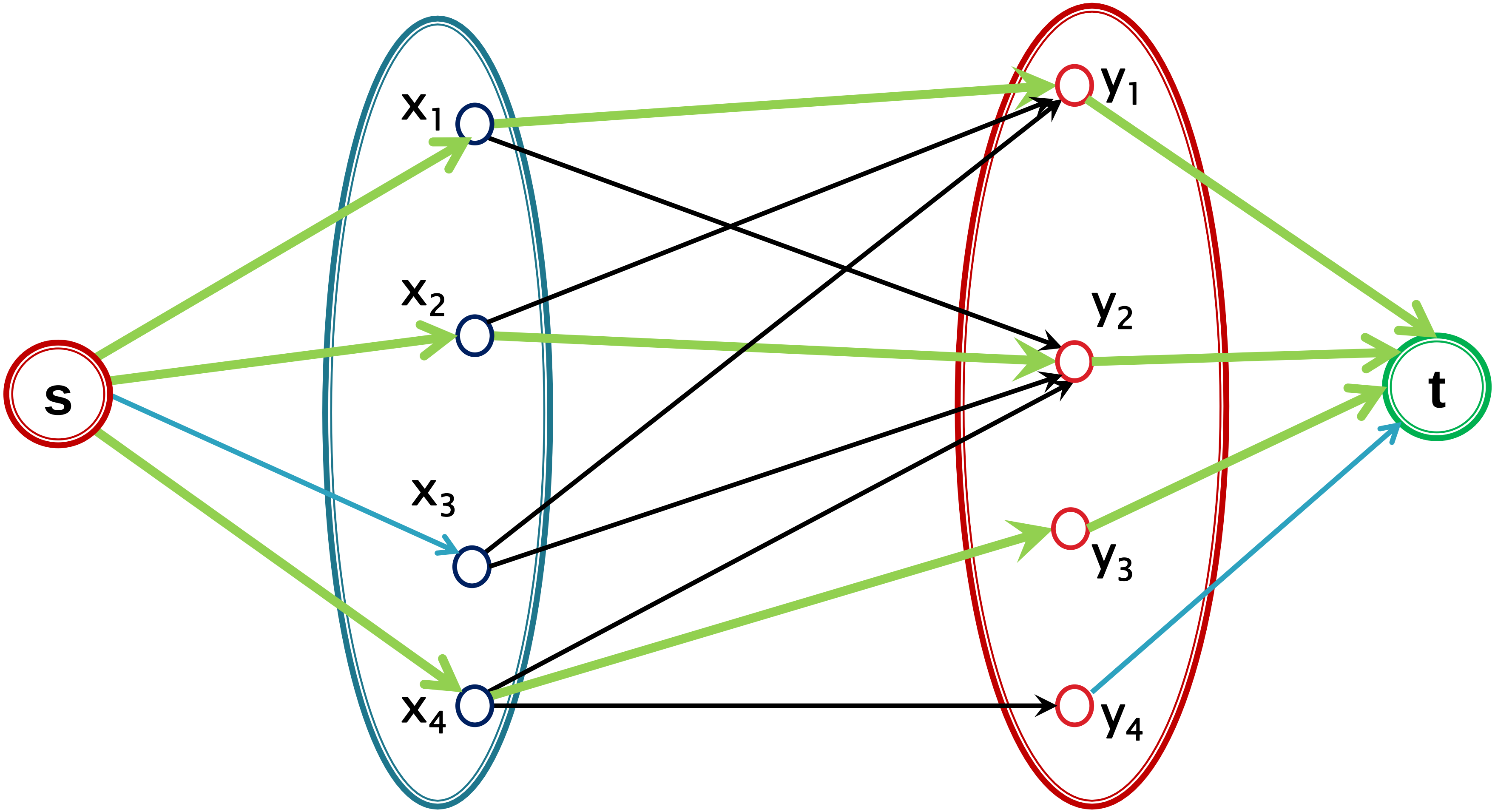
$$\mathbf{val(f) = |M|}$$

Justificare. **Dat un cuplaj M în G, se poate construi un flux f în N_G cu val(f) = |M| astfel:**

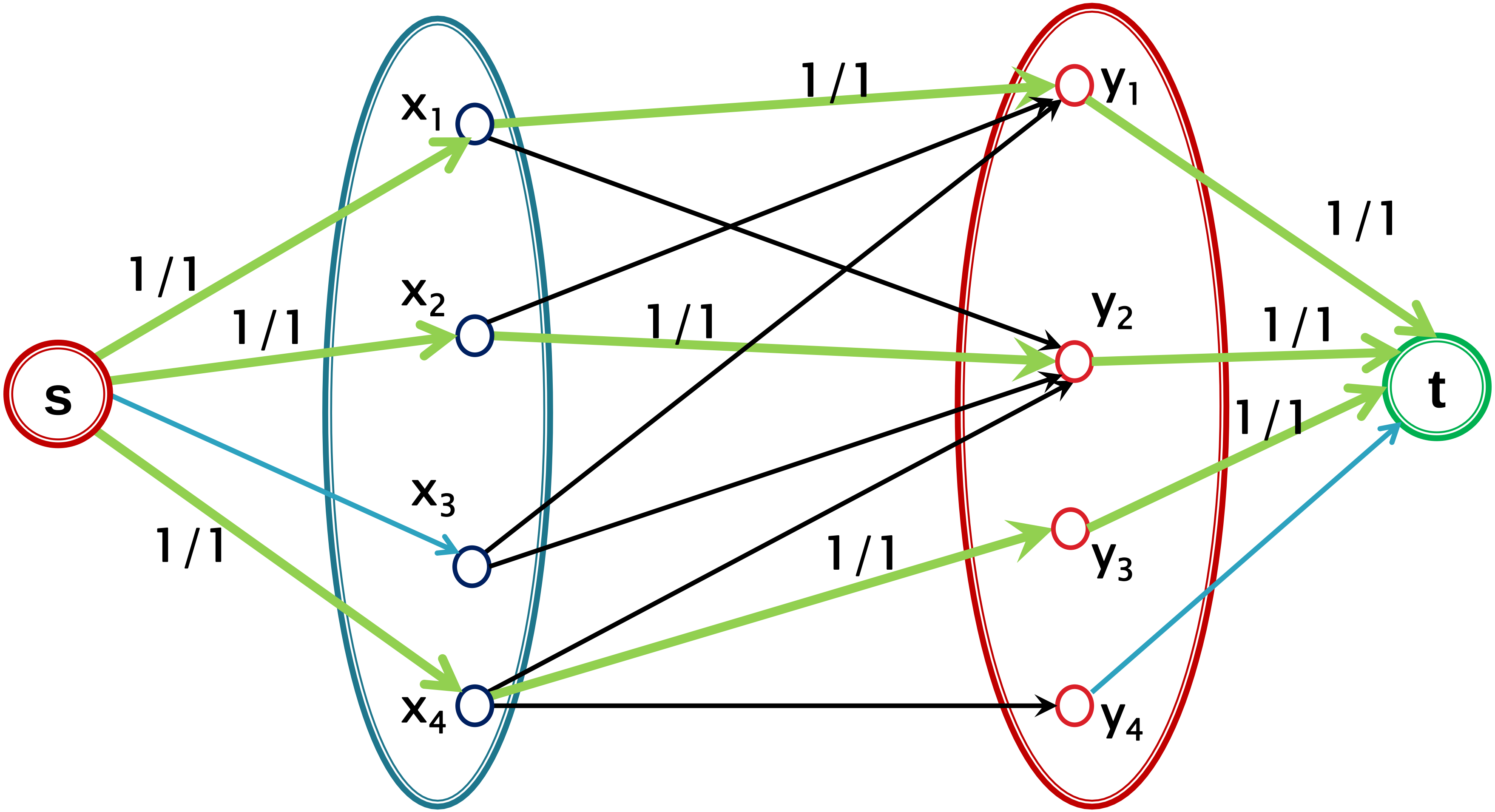
- Dat cuplaj în graf $\Rightarrow$ definim flux în reţea



- Dat cuplaj în graf $\Rightarrow$ definim flux în reţea

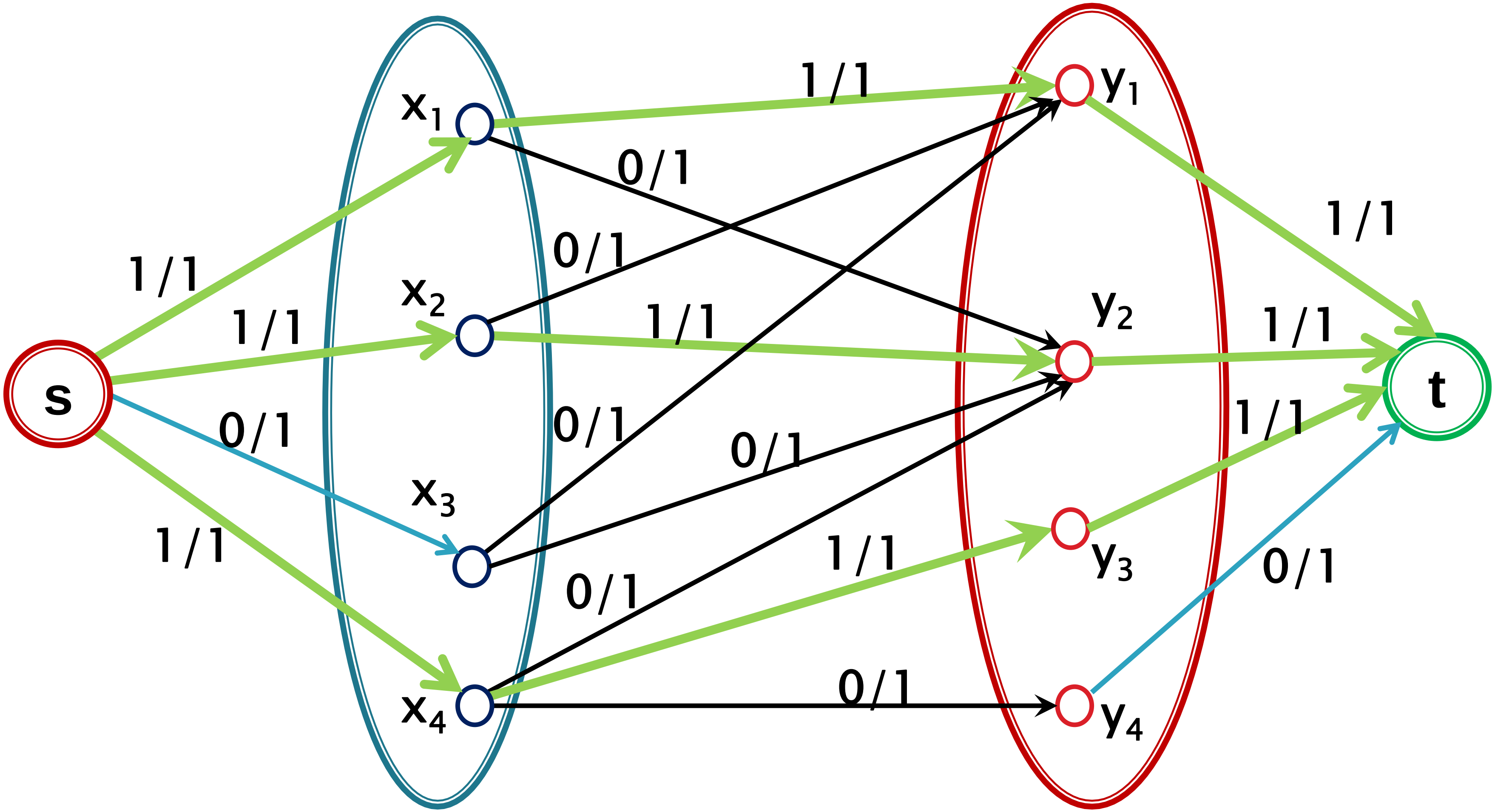


- Dat cuplaj în graf $\Rightarrow$ definim flux în reţea



Celelalte arce au flux 0 și capacitate 1

- Dat cuplaj în graf $\Rightarrow$ definim flux în reţea



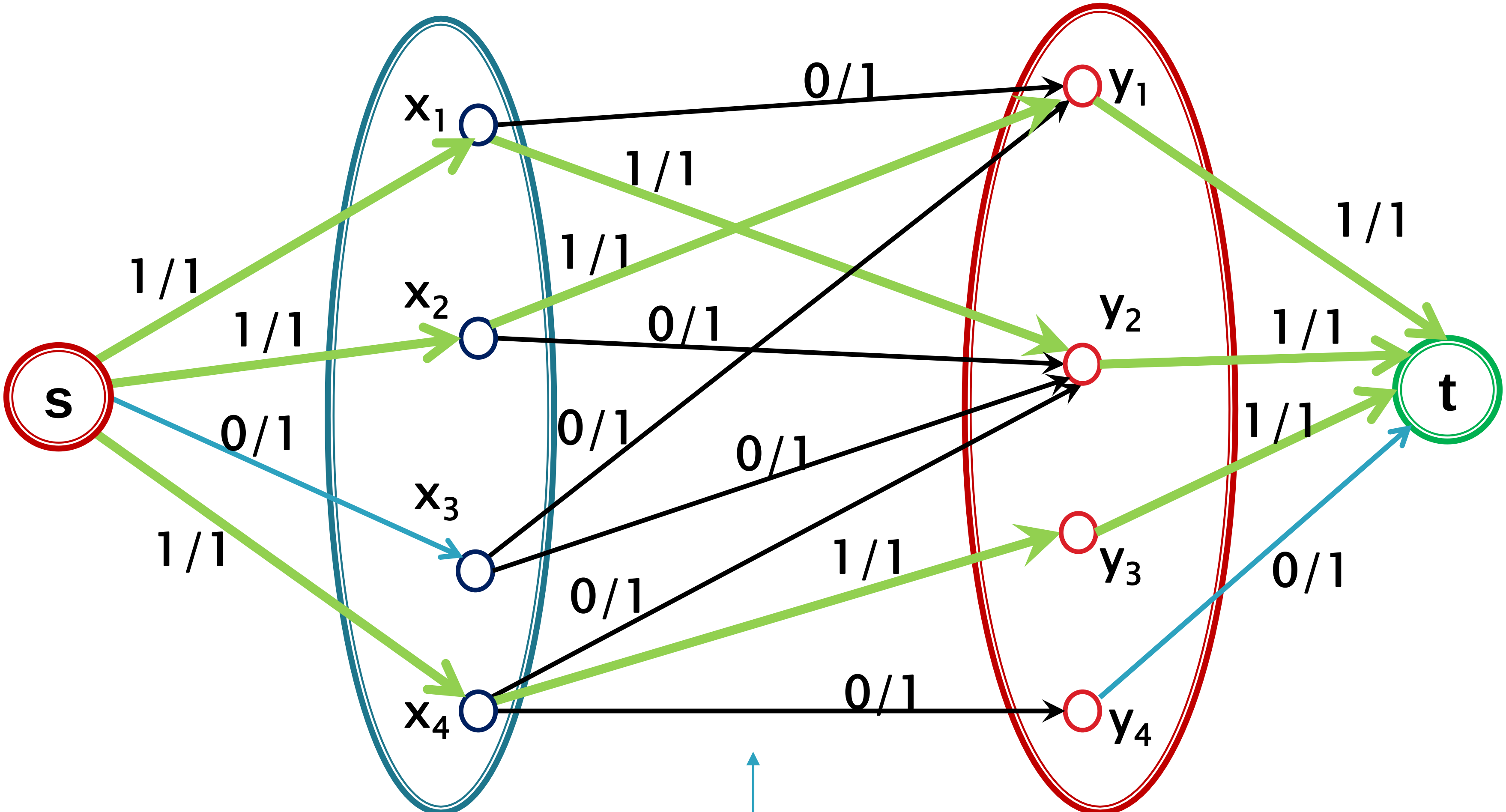
Proprietatea 2

Fie $G=(X \cup Y, E)$ un graf bipartit şi f un flux în reţeaua de transport N_G asociată. Atunci există M un cuplaj în G cu

$$val(f) = |M|$$

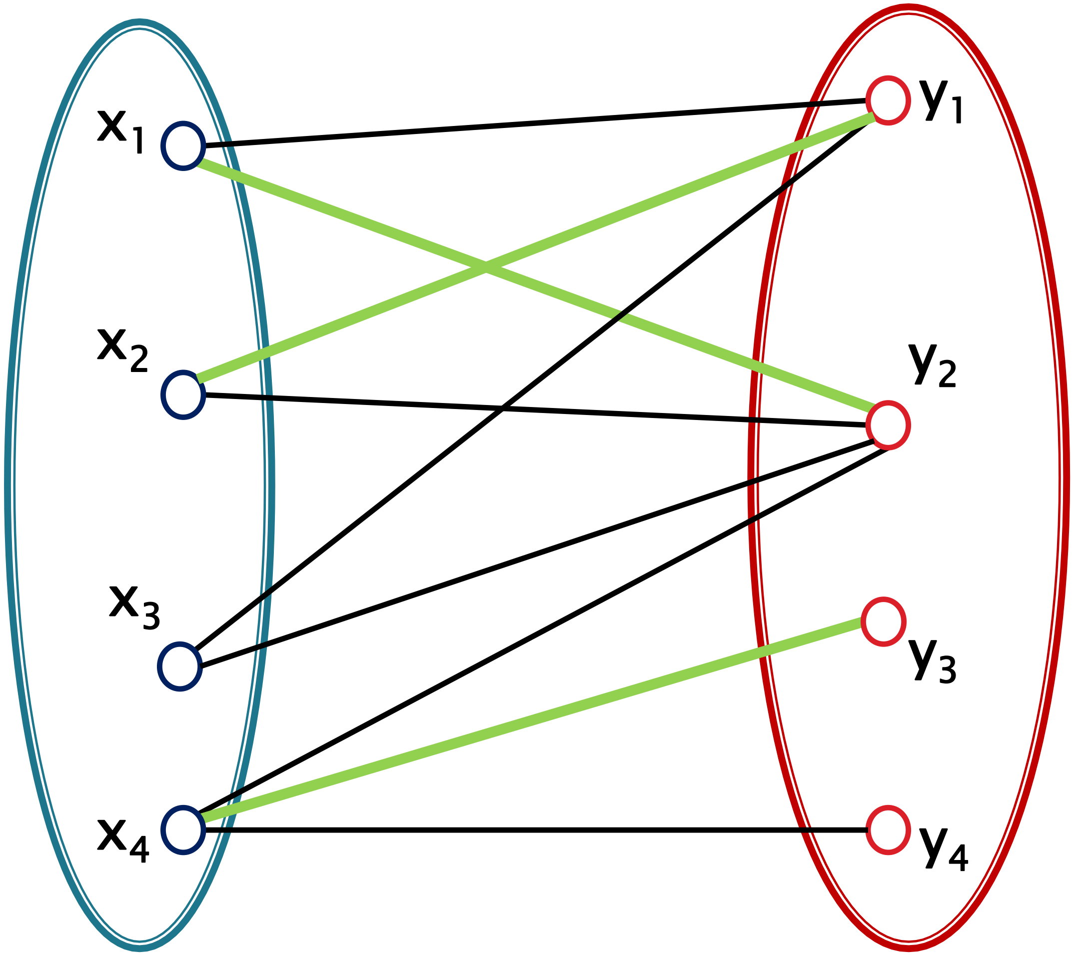
Justificare. **Dat un flux f în N, se poate construi un cuplaj M în G cu val(f) = |M| astfel:**

- Dat flux în reţea ⇒ cuplaj în graf



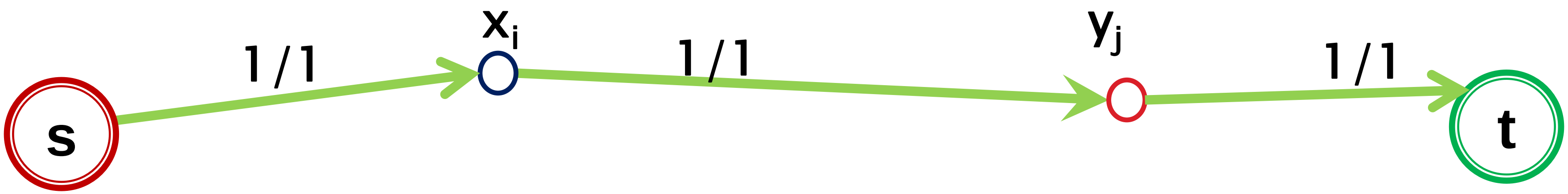
arcele care au flux pozitiv dau muchiile din M

- Dat flux în reţea ⇒ cuplaj în graf

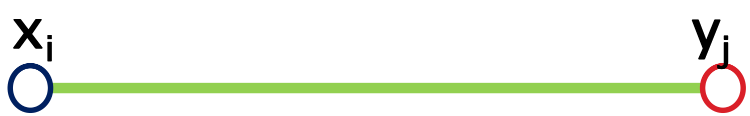


- Concluzie: Flux în reţea $\Leftrightarrow$ cuplaj în graf

Drum cu o unitate de flux



Muchie în cuplaj



Consecinţă

f^* flux maxim în N $\Rightarrow$ cuplajul corespunzător M^* este cuplaj maxim în G

A determina un **cuplaj maxim** într-un graf bipartit $\Leftrightarrow$
a determina un **flux maxim** în reţeaua asociată

Algoritm de determinare a unui cuplaj maxim în $G=(X\cup Y,E)$:

1. **Construim N reţeaua de transport asociată**

2. **Determinăm f^* flux maxim în N**

3. **Considerăm $M = \{xy \mid f^*(xy)=1,\ x\in X,\ y\in Y,\ xy\in N\}$**

(pentru fiecare arc cu flux nenul xy din N care nu este incident în s sau t, muchia xy corespunzătoare din G se adaugă la M)

4. **return M**

Algoritm de determinare a unui cuplaj maxim în $G=(X\cup Y,E)$:

1. **Construim N rețeaua de transport asociată**
2. **Determinăm f^* flux maxim în N**
3. **Considerăm $M = \{xy \mid f^*(xy)=1,\ x\in X,\ y\in Y,\ xy\in N\}$**

   (pentru fiecare arc cu flux nenul xy din N care nu este incident în s sau t, muchia xy corespunzătoare din G se adaugă la M)

4. **return M**

Complexitate?

Algoritm de determinare a unui cuplaj maxim în $G=(X\cup Y,E)$:

1. **Construim N reţeaua de transport asociată**

2. **Determinăm f^* flux maxim în N**

3. **Considerăm $M = \{xy \mid f^*(xy)=1,\ x\in X,\ y\in Y,\ xy\in N\}$**

(pentru fiecare arc cu flux nenul xy din N care nu este incident în s sau t, muchia xy corespunzătoare din G se adaugă la M)

4. **return M**

Complexitate: $C=1$ (sau $L \le c^+(s) \le n$) $\Rightarrow$ $O(mn)$

Aplicaţie
Construcţia unui graf orientat din secvenţele de grade

- Se dau secvenţele
  - $s_0^+ = \{d_1^+, \ldots, d_n^+\}$
  - $s_0^- = \{d_1^-, \ldots, d_n^-\}$

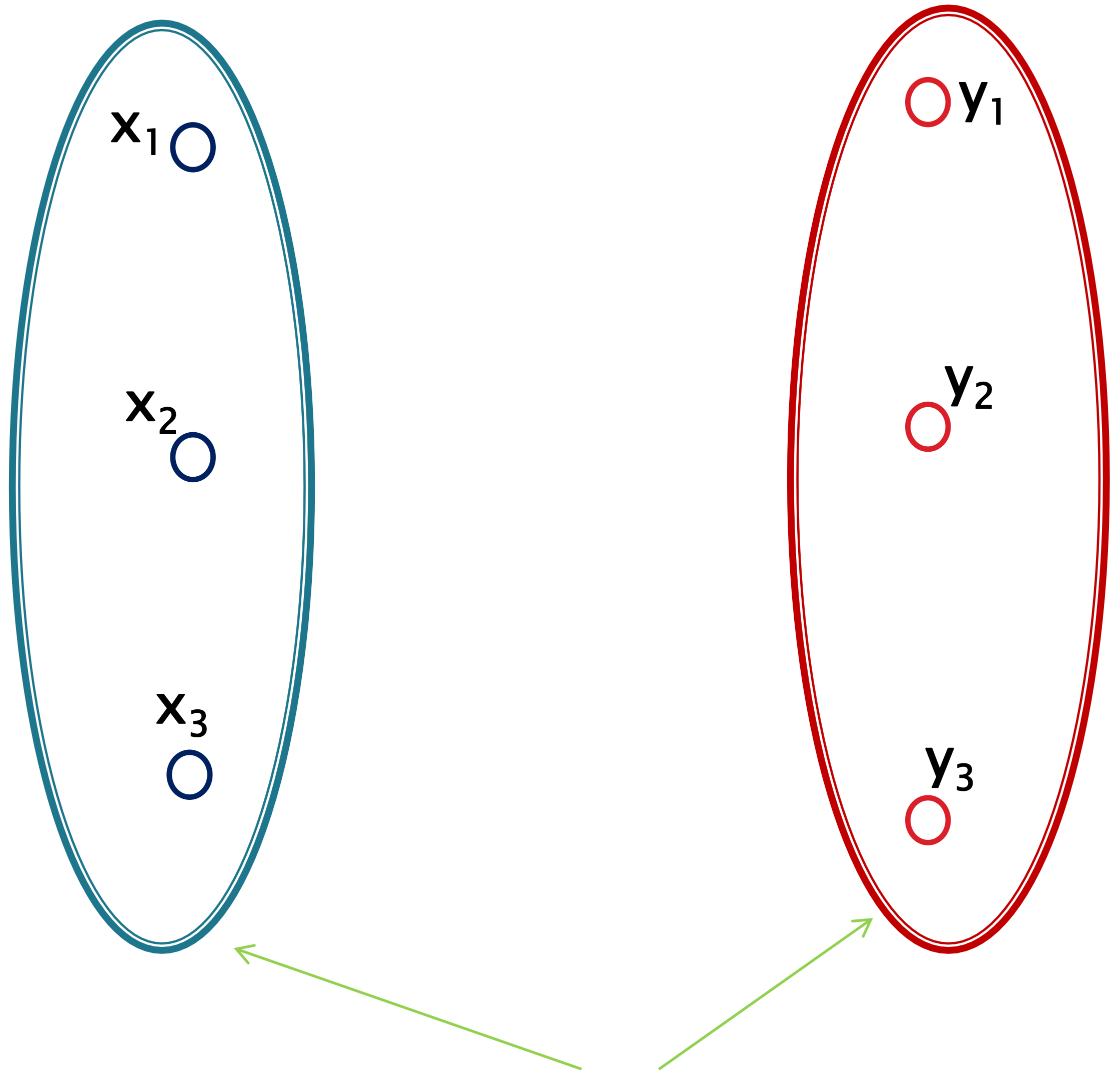
  cu $d_1^+ + \ldots + d_n^+ = d_1^- + \ldots + d_n^-$

Să se construiască, **dacă se poate**, un graf orientat G cu $s^+(G) = s_0^+$ şi $s^-(G) = s_0^-$

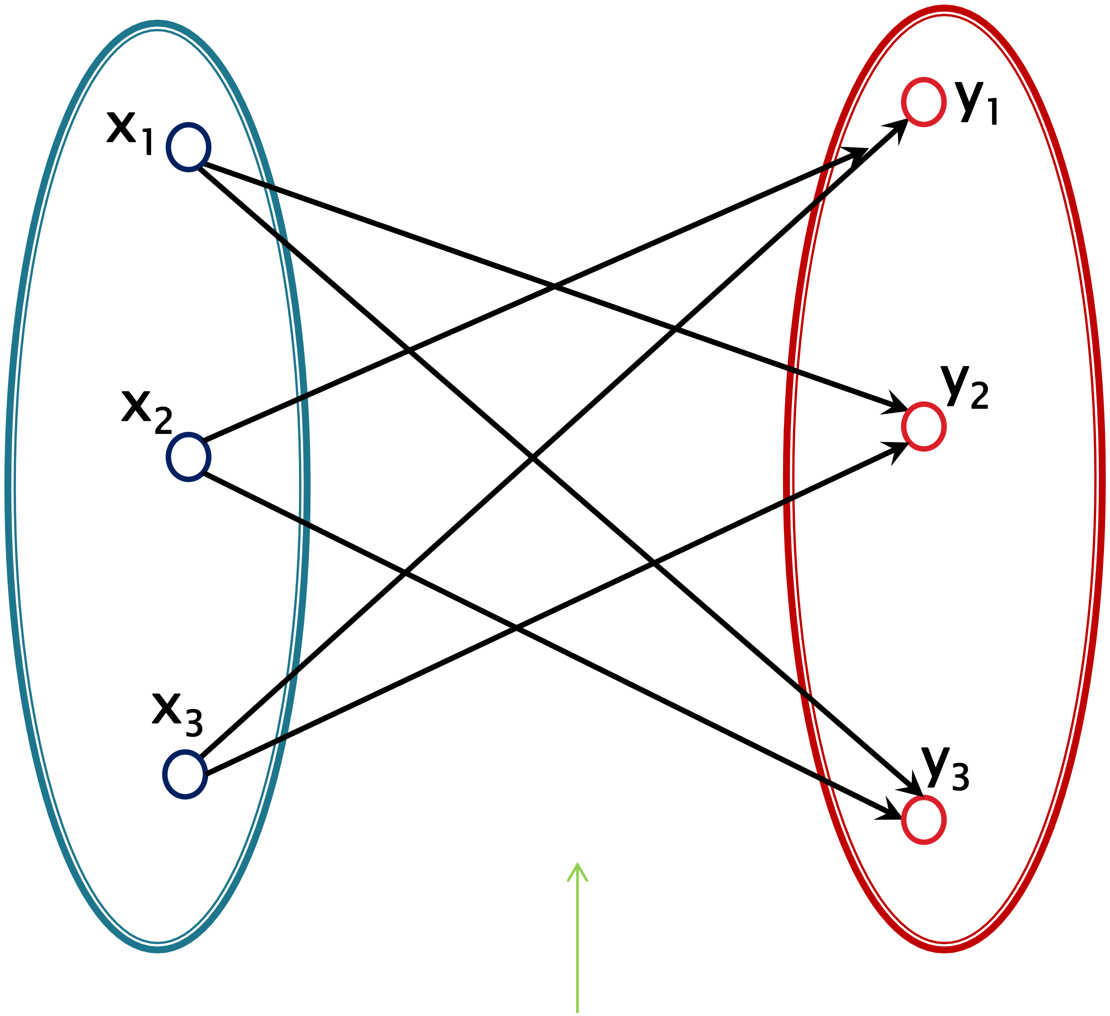
- **Exemplu**
  - $s_0^+ = \{1, 0, 2\}$
  - $s_0^- = \{1, 1, 1\}$

- **Exemplu**
  - $s_0^+ = \{1, 0, 2\}$
  - $s_0^- = \{1, 1, 1\}$

- **Construim o reţea asociată celor două secvenţe** a.î. din fluxul maxim în reţea să putem **deduce** dacă G se poate construi + arcele grafului G (în caz afirmativ)

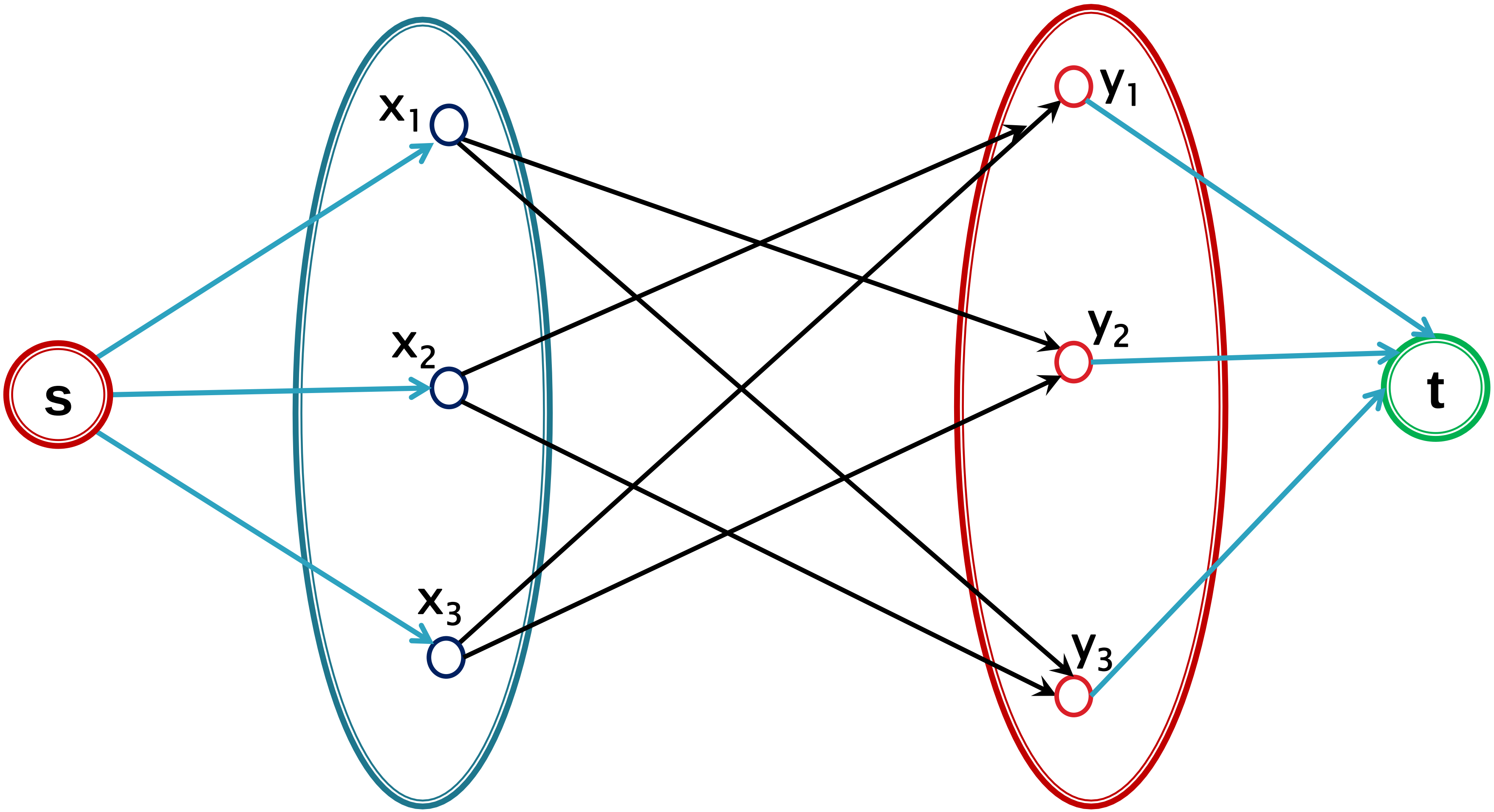


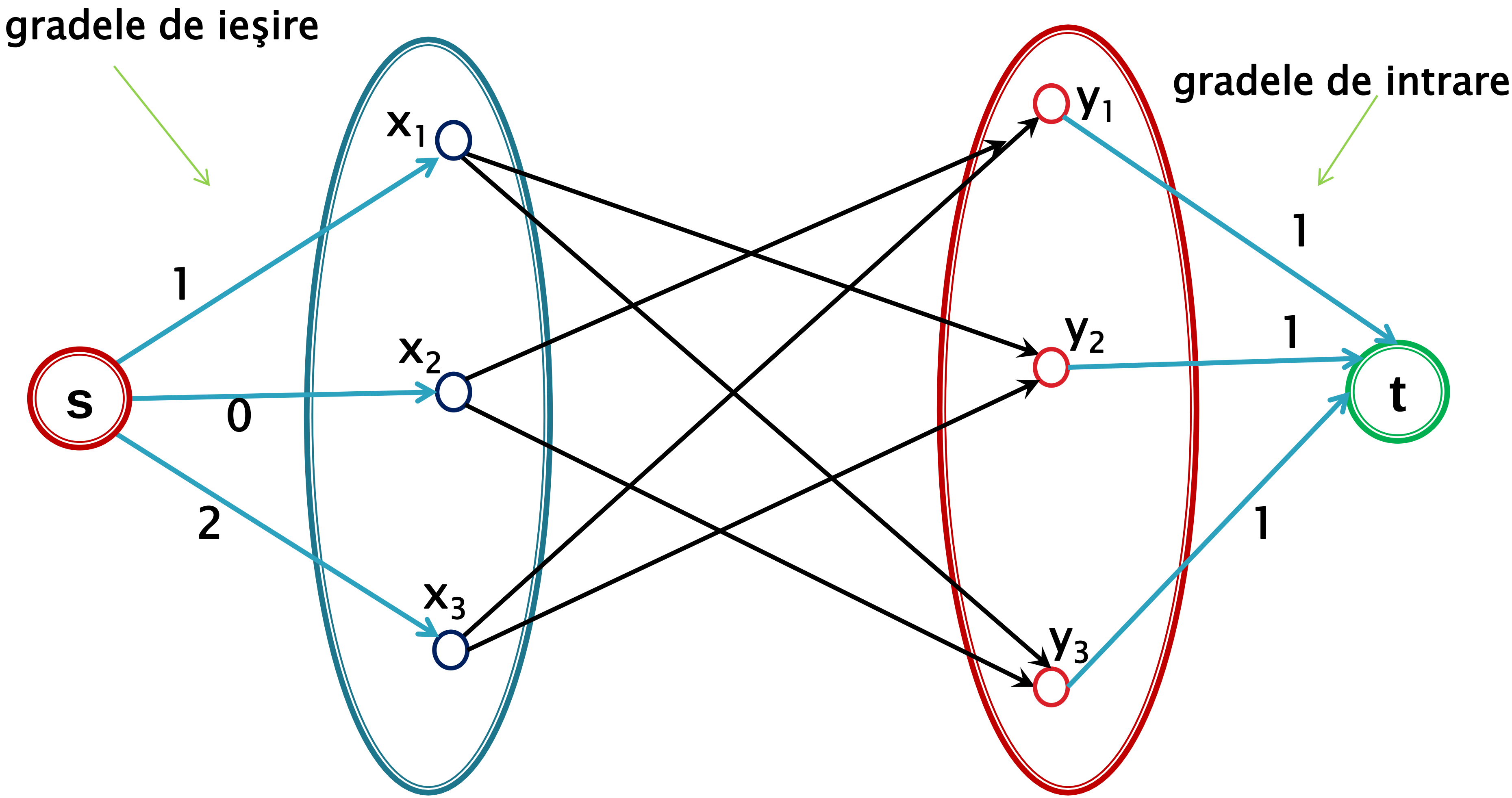
Vârfurile 1, 2,..., n se pun în ambele clase ale bipartiţiei (câte o copie)

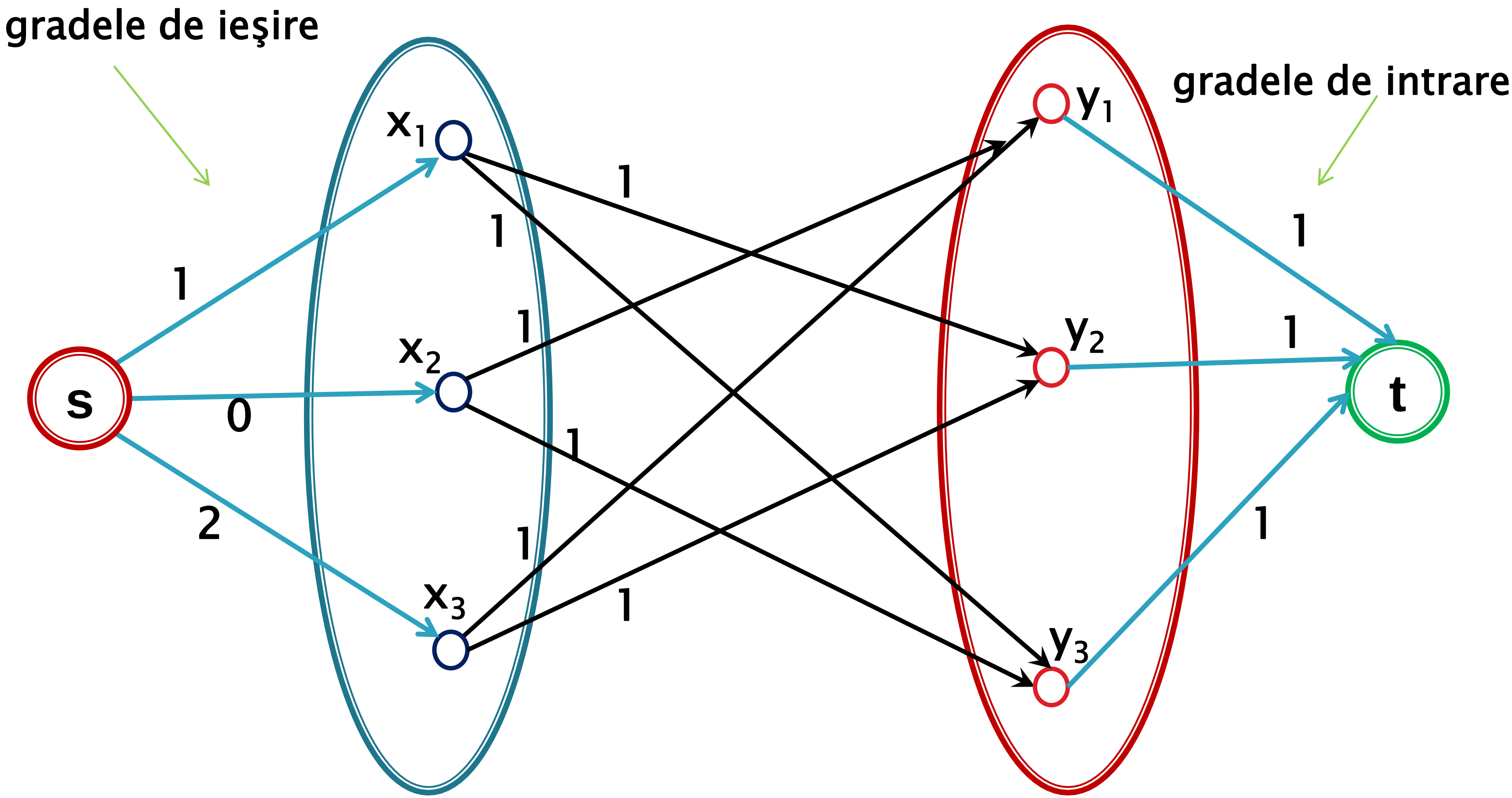


arce $x_i y_j$ cu $i \neq j$

(fluxul pe arcul $x_i y_j$ va fi nenul $\Leftrightarrow$ $ij \in E(G)$)


s
x_1
x_2
x_3
y_1
y_2
y_3
t


gradele de ieşire
gradele de intrare
s
t
x_1
x_2
x_3
y_1
y_2
y_3
1
0
2
1
1
1


gradele de ieşire
gradele de intrare
s
t
x_1
x_2
x_3
y_1
y_2
y_3
1
0
2
1
1
1
1
1
1
1
1
1

- **Proprietate**

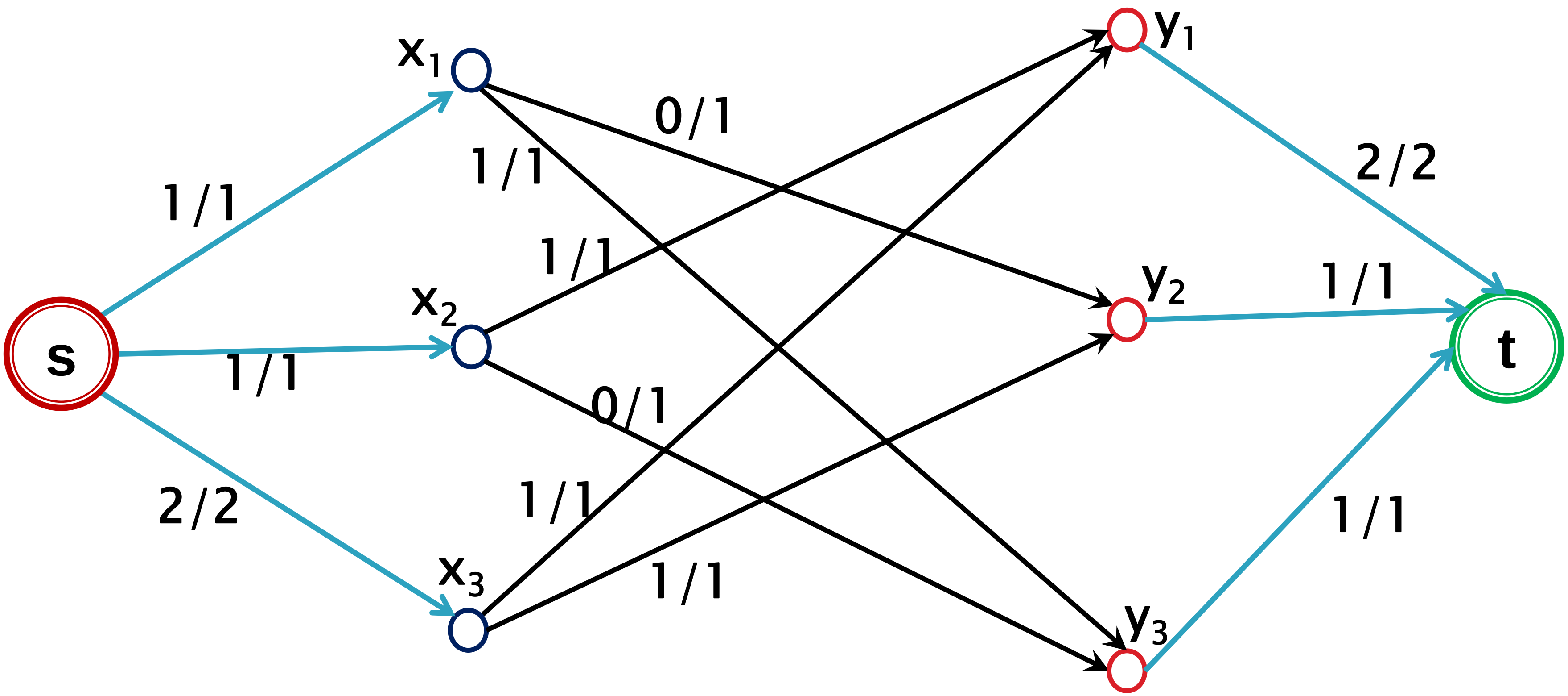
  Există graf cu secvenţele date $\Leftrightarrow$ în rețeaua asociată **fluxul de valoare maximă** are

$$\text{val(f)} = d_1^+ + \ldots + d_n^+ = d_1^- + \ldots + d_n^-$$

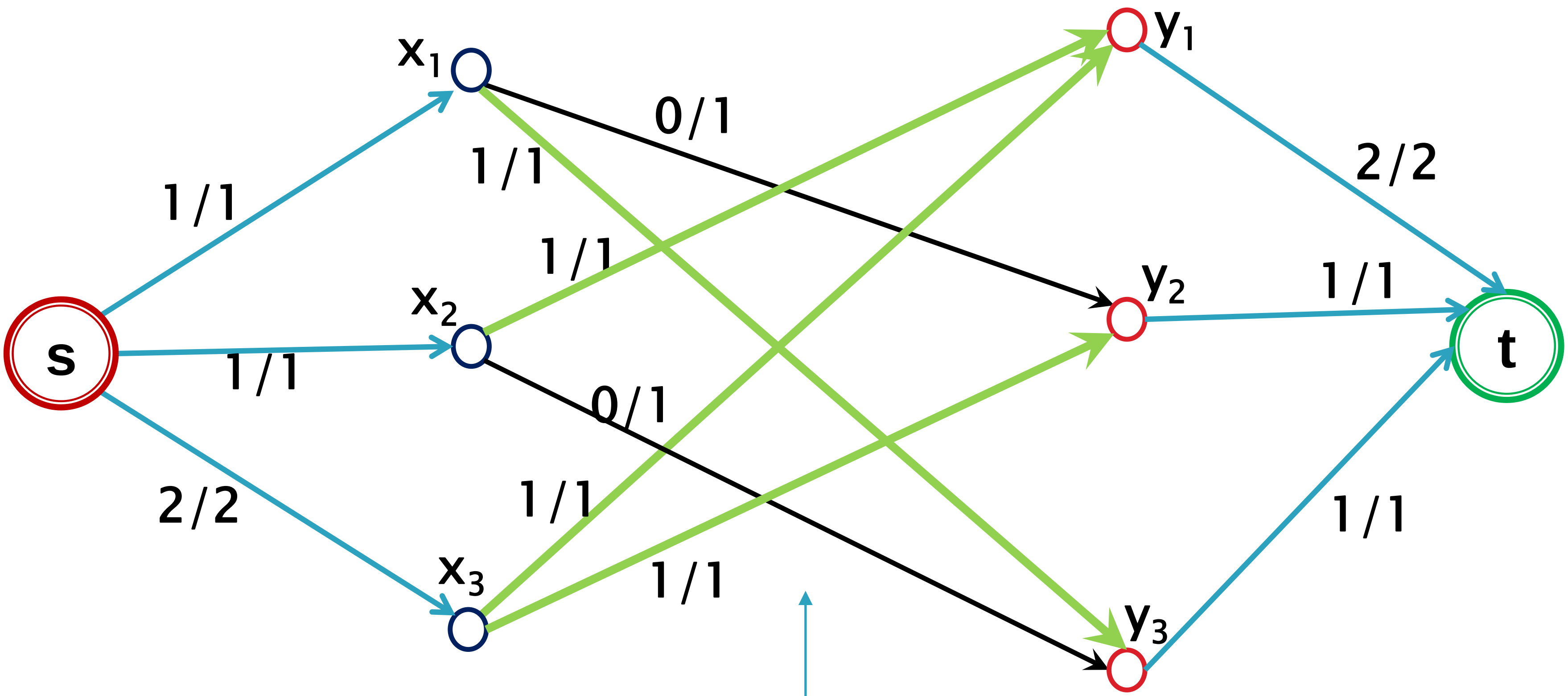
(**saturează toate arcele care ies din s și toate arcele care intră în t**)

tăieturile ({s},V–{s}) , (V–{t}, {t}) sunt minime

flux în reţea care saturează arcele din s și t ⇒ G

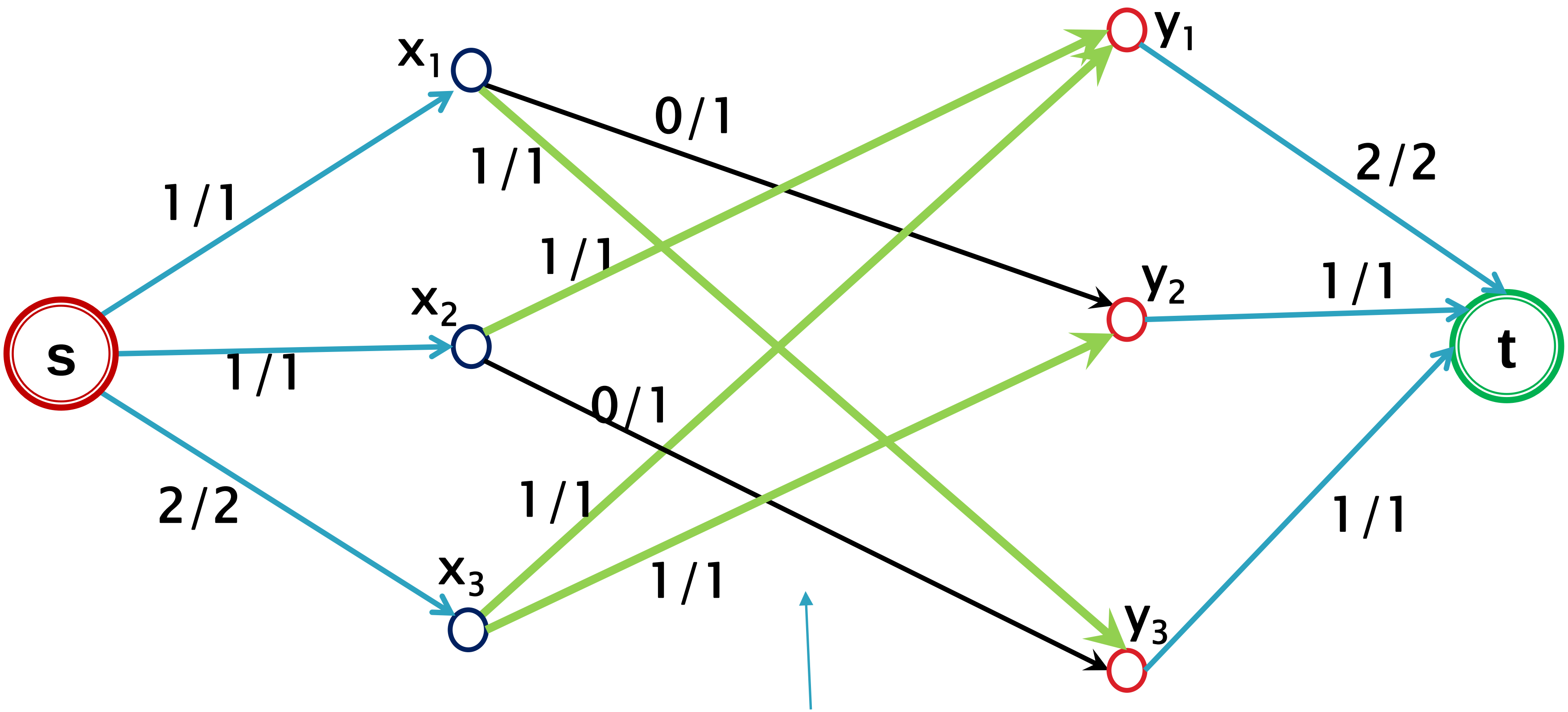


flux în reţea care saturează arcele din s şi t ⇒ G



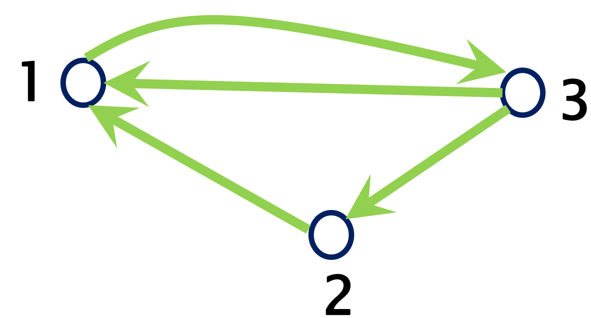
arcele care au flux pozitiv dau arcele lui G

flux în reţea care saturează arcele din s şi t ⇒ G



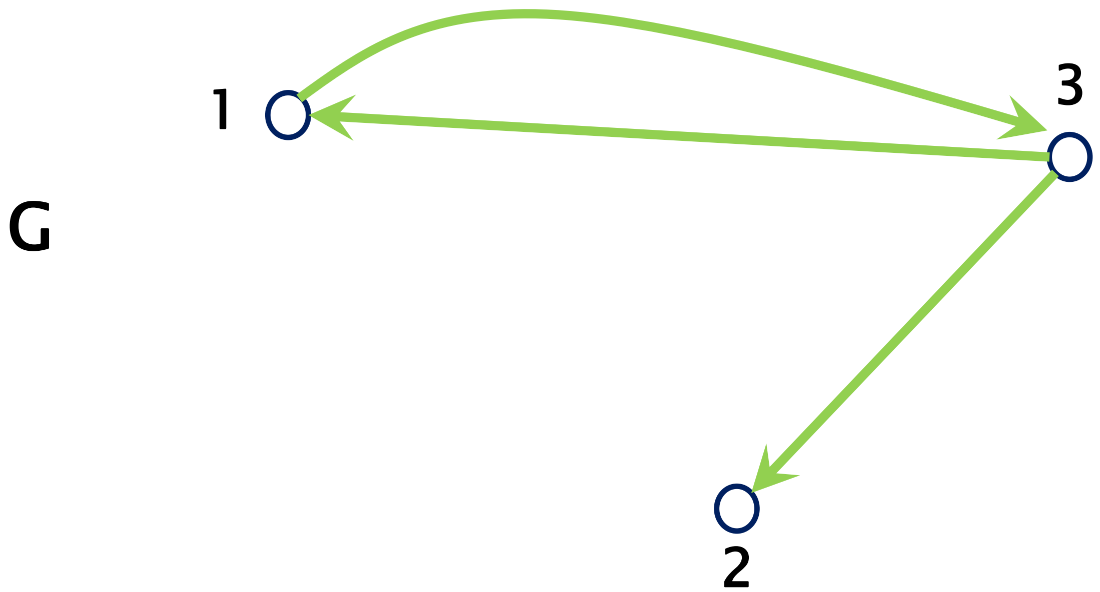
arcele care au flux pozitiv dau arcele lui G

- x_1y_3 => arcul (1, 3)
- x_2y_1 => arcul (2, 1)
- x_3y_1 => arcul (3, 1)
- x_3y_2 => arcul (3, 2)



Reciproc

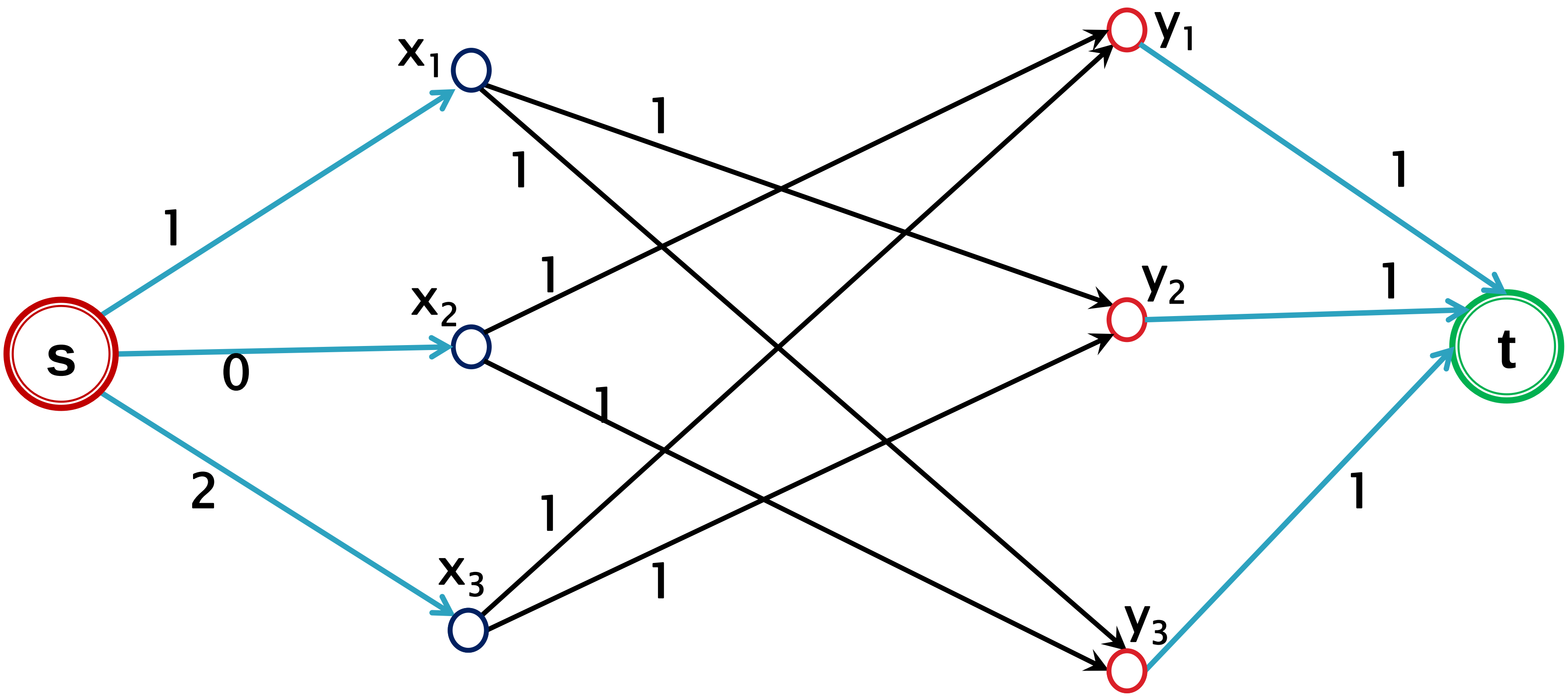
G ⇒ flux în reţea

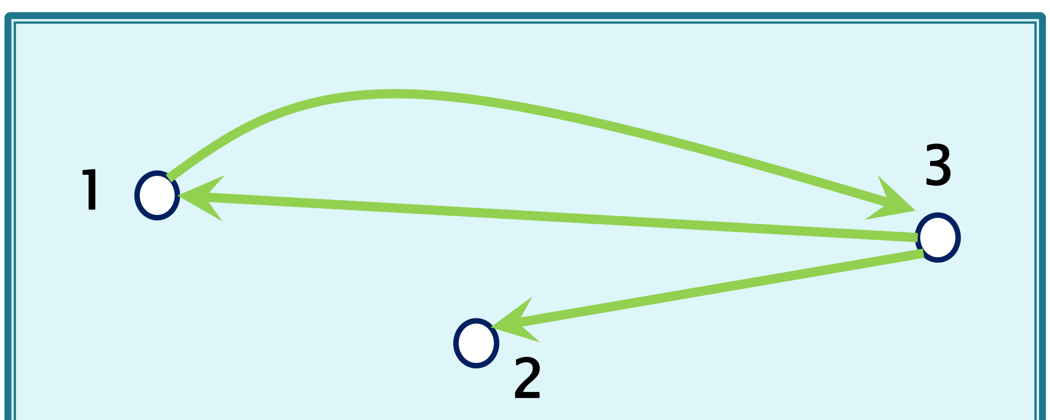


$$s_0^+ = \{1, 0, 2\}$$

$$s_0^- = \{1, 1, 1\}$$

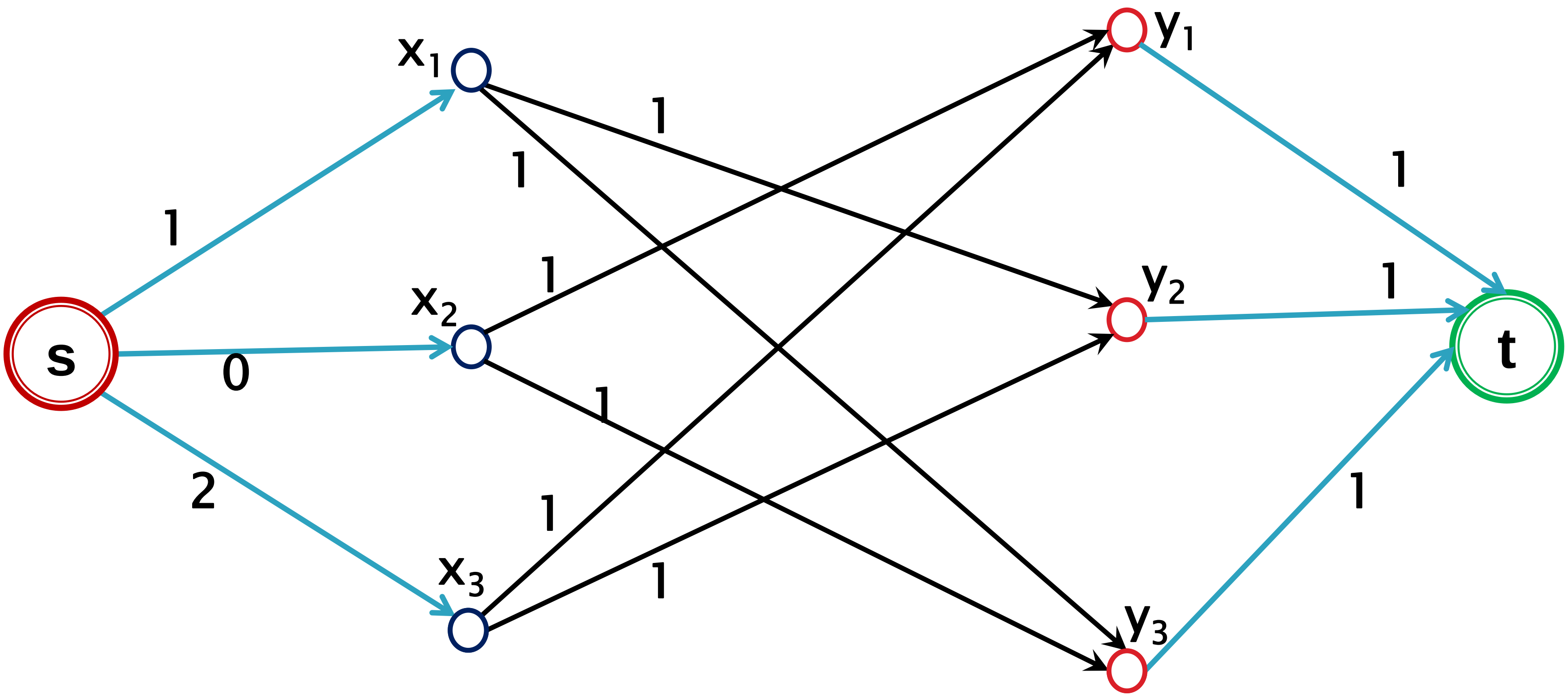
G ⇒ flux în reţea





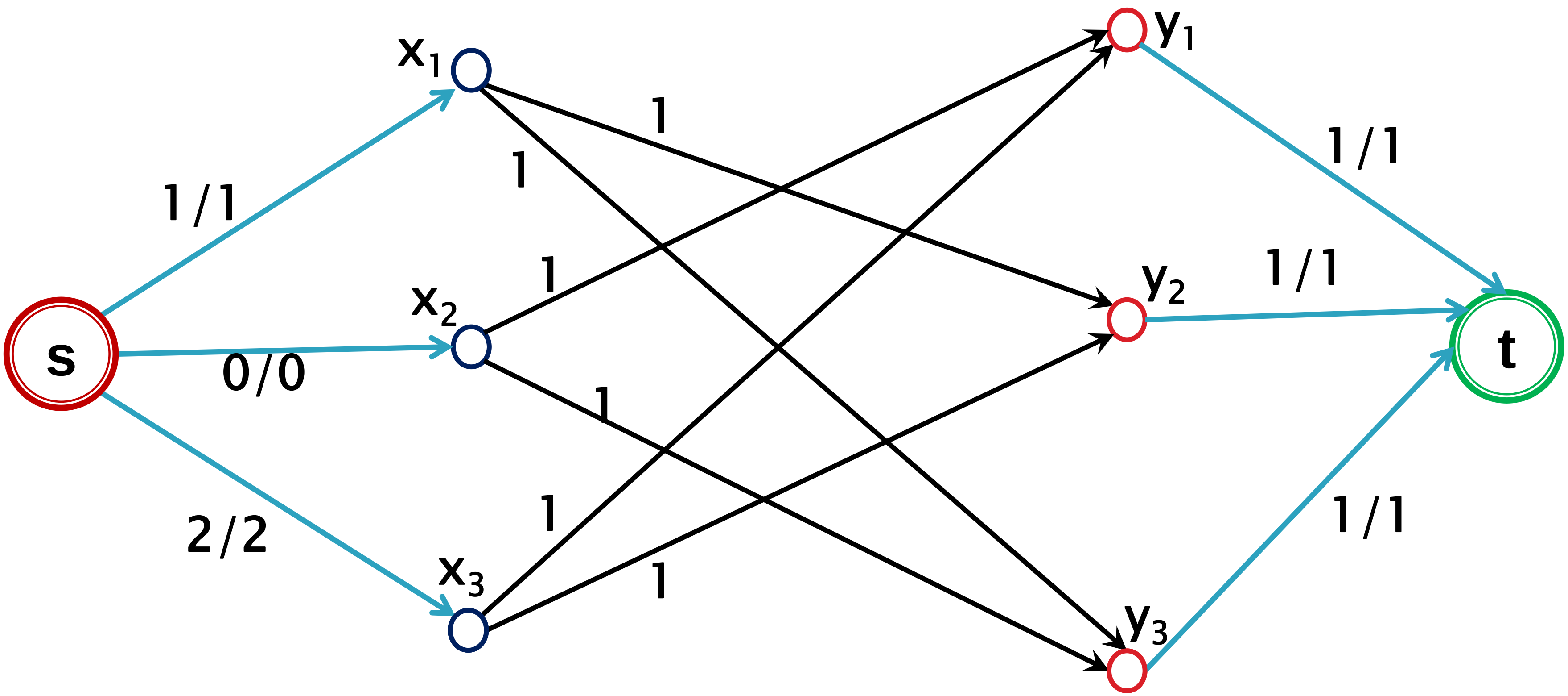
G

G ⇒ flux în reţea

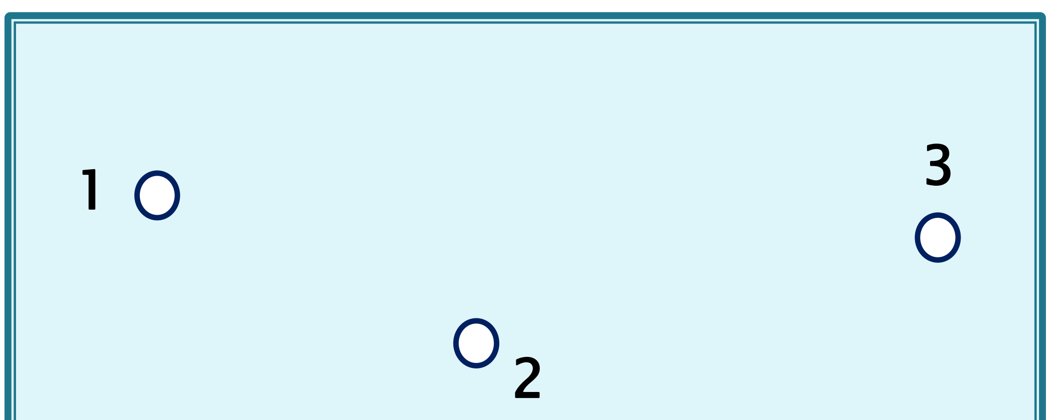


saturăm arcele
din s și t

G ⇒ flux în reţea

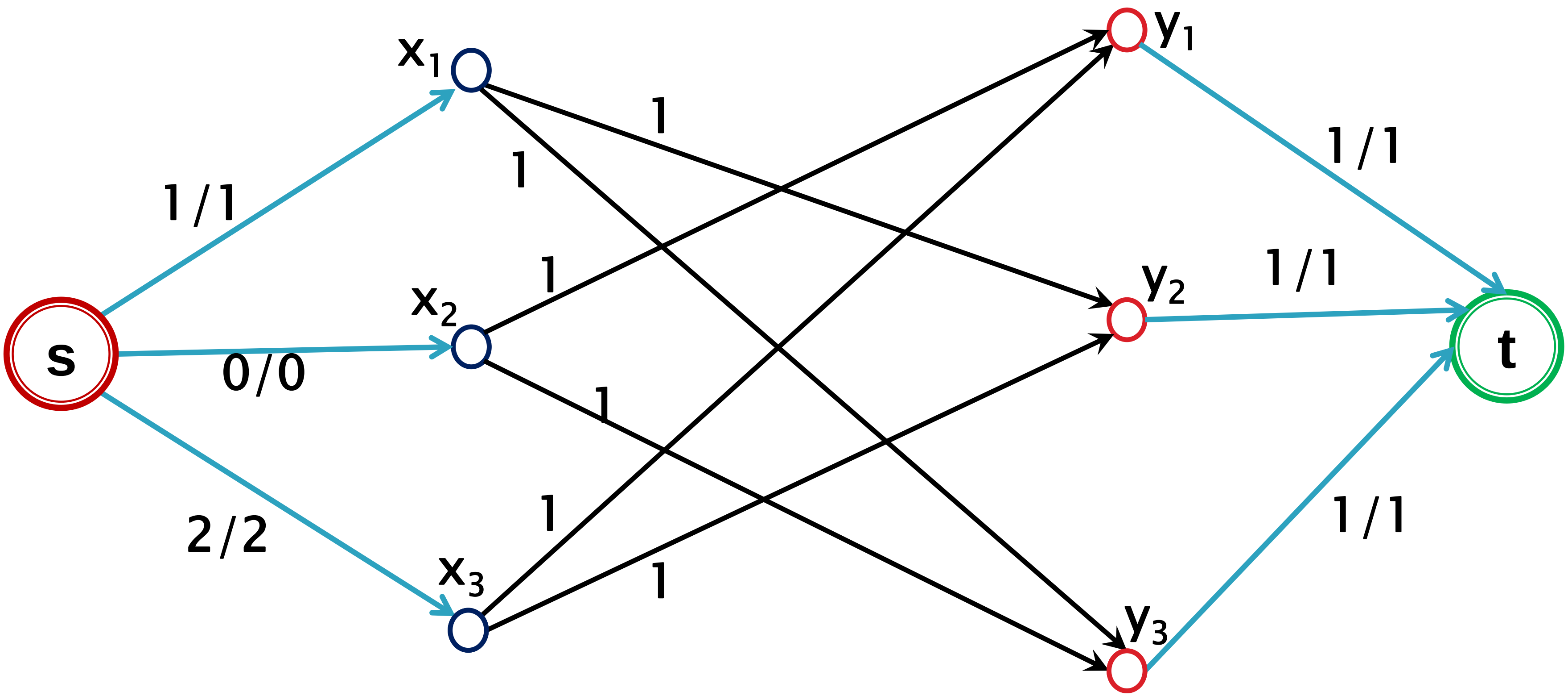


asociem flux
arcelor din G

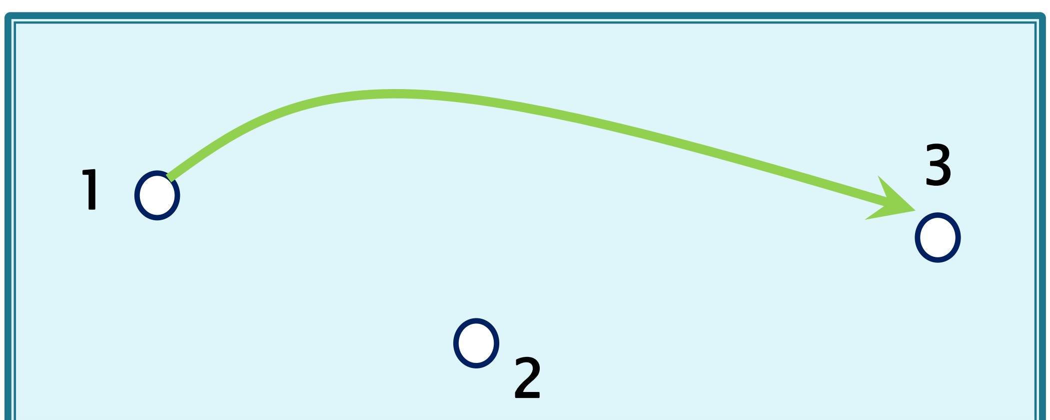


G

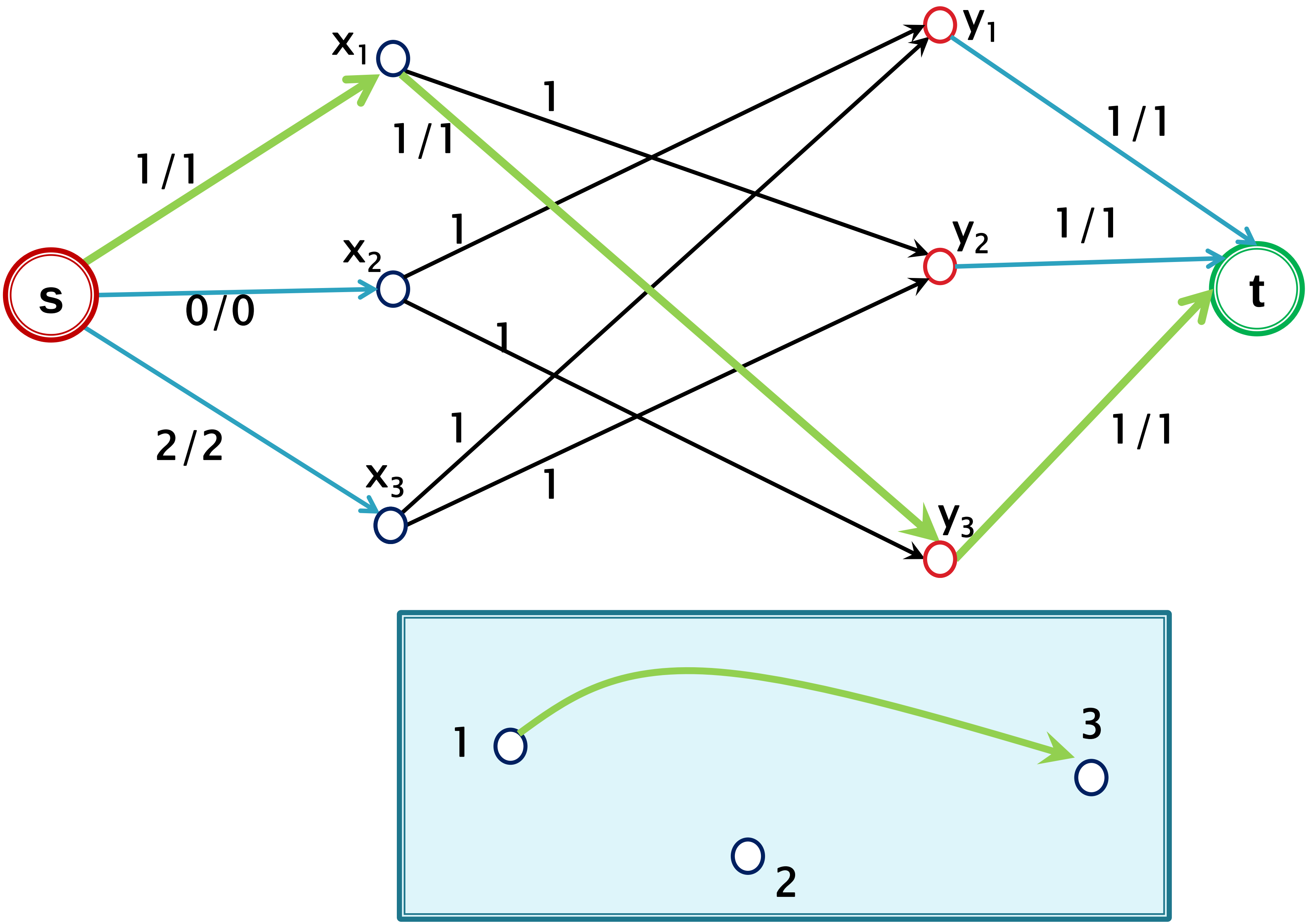
G ⇒ flux în reţea



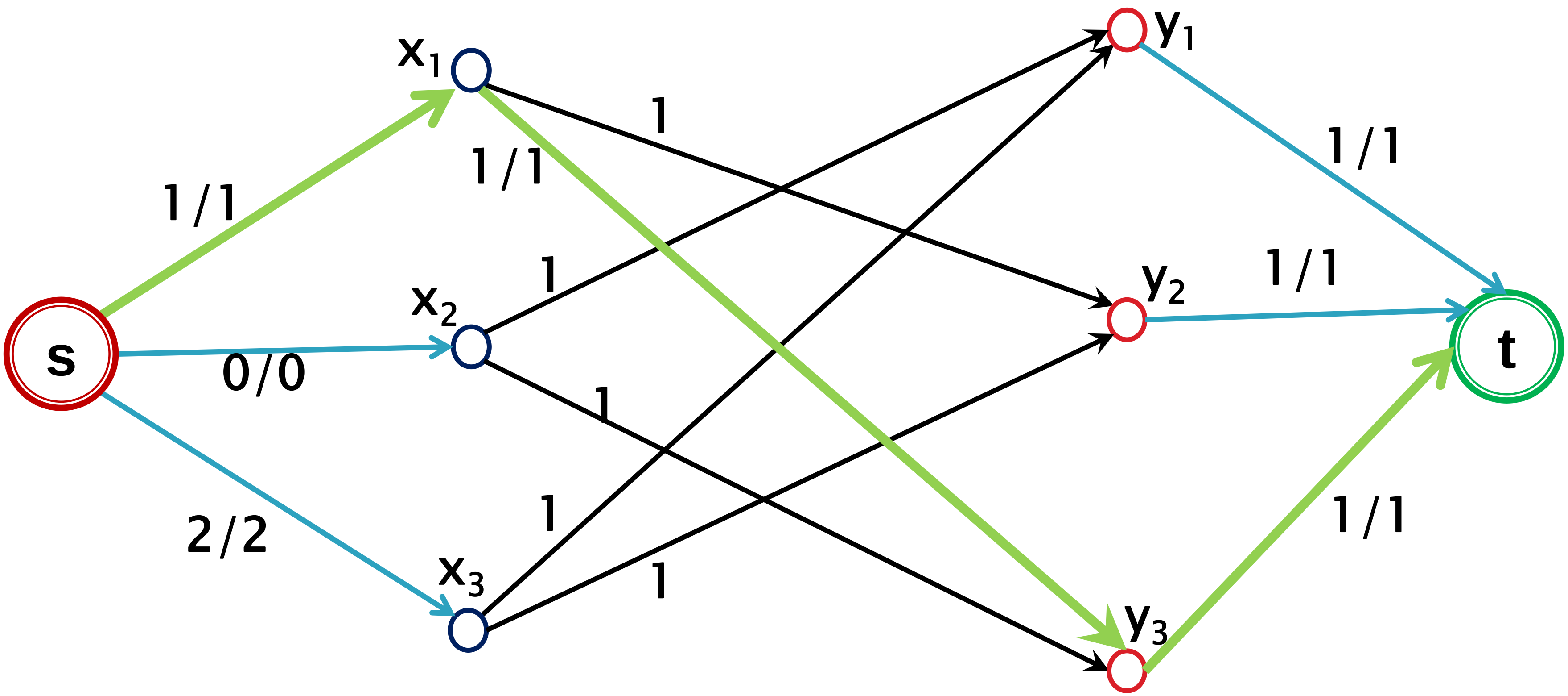
Arc (1,3) =>
flux pe arcul x_1y_3



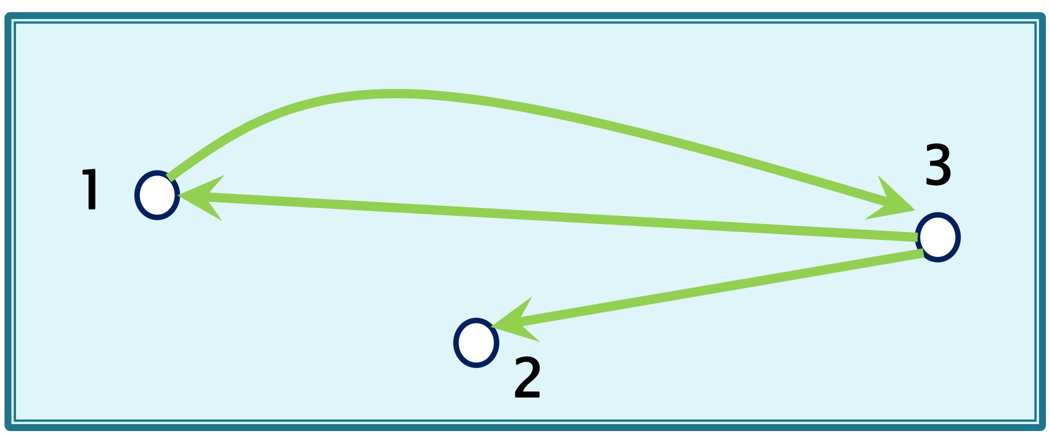
G ⇒ flux în reţea



G ⇒ flux în reţea

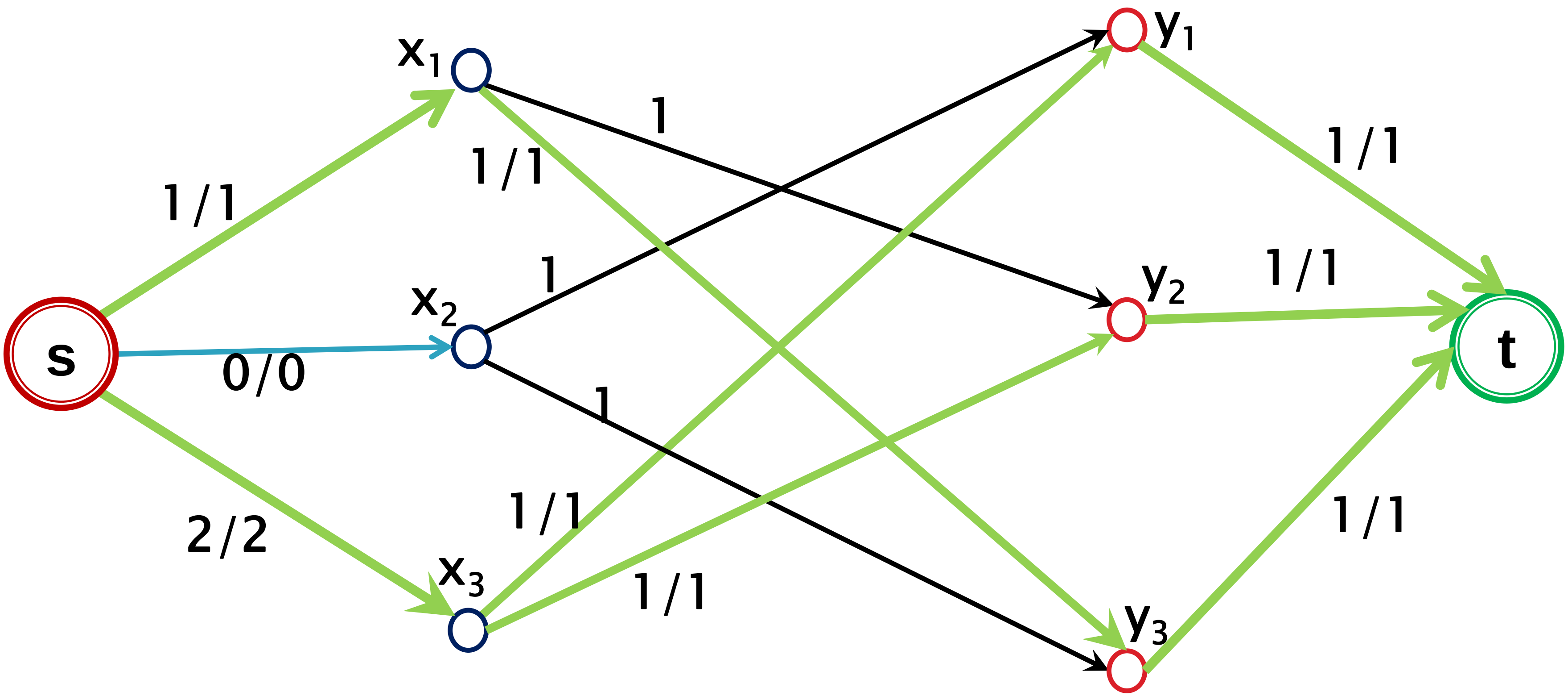


Arce (3,1), (3,2)
=> ?



G

G ⇒ flux în reţea



Restul arcelor au fluxul 0

Algoritm de determinare a unui graf orientat G cu

$s^+(G) = s_0^+$ **şi** $s^-(G) = s_0^-$

1. **Construim N reţeaua de transport asociată**

2. **Determinăm f^* flux maxim în N**

Algoritm de determinare a unui graf orientat G cu

s$^+$(G) $= s_0^+$ şi s$^-$(G) $= s_0^-$

1. **Construim N reţeaua de transport asociată**
2. **Determinăm f* flux maxim în N**
3. **Dacă val(f*)< $d_1^+ + \ldots + d_n^+$ atunci**

   Nu există G. STOP

Algoritm de determinare a unui graf orientat G cu $s^+(G) = s_0^+$ şi $s^-(G) = s_0^-$

1. Construim N reţeaua de transport asociată
2. Determinăm f^* flux maxim în N
3. Dacă val(f*)< $d_1^+ + \ldots + d_n^+$ atunci

   Nu există G. STOP

4. V(G) = {1,…, n}

   E(G) = {ij| $x_i y_j \in$ N cu $f^*(x_i y_j)$ = 1}

Complexitate: $L \leq c^+(s) = d_1^+ + \ldots + d_n^+ = m \Rightarrow O(m^2)$

Aplicaţii
Alte probleme de asociere

Probleme de asociere (temă)

- Se dau 2 mulțimi de obiecte, spre exemplu produse (aflate în fabrici) și clienți (joburi/masini, pagini web/servere, echipe turneu etc).

- Pentru fiecare produs x se cunoaște

c(x) = numărul de unități disponibile din produsul x

- Pentru fiecare client y se cunoaște

c(y) = numărul maxim de unități de produse pe care le poate primi (în total, din toate produsele)

- Pentru fiecare pereche produs-client (x,y) se cunoaște

c(x,y) = numărul maxim de unități din produsul x pe care le poate primi clientul y

Să se determine o modalitate de a distribui cât mai multe produse (unităţi de produse) clienților cu respectarea constrângerilor

Probleme de asociere (temă)

- **Observație** – Problema determinării unui cuplaj de cardinal maxim într-un graf bipartit $G=(X \cup Y,E)$ este un caz particular al acestei probleme, pentru

  – $c(x) = c(y) = 1, \forall\ x \in X, y \in Y$

  – $c(x,y) = 1$, dacă $xy \in E$

  – $c(x,y) = 0$, dacă $xy \notin E$

  - http://jeffe.cs.illinois.edu/teaching/algorithms/2009/notes/17-maxflowapps.pdf

Aplicaţie

Drumuri arc-disjuncte între două vârfuri.

Conectivitatea unui graf

(SUPLIMENTAR)

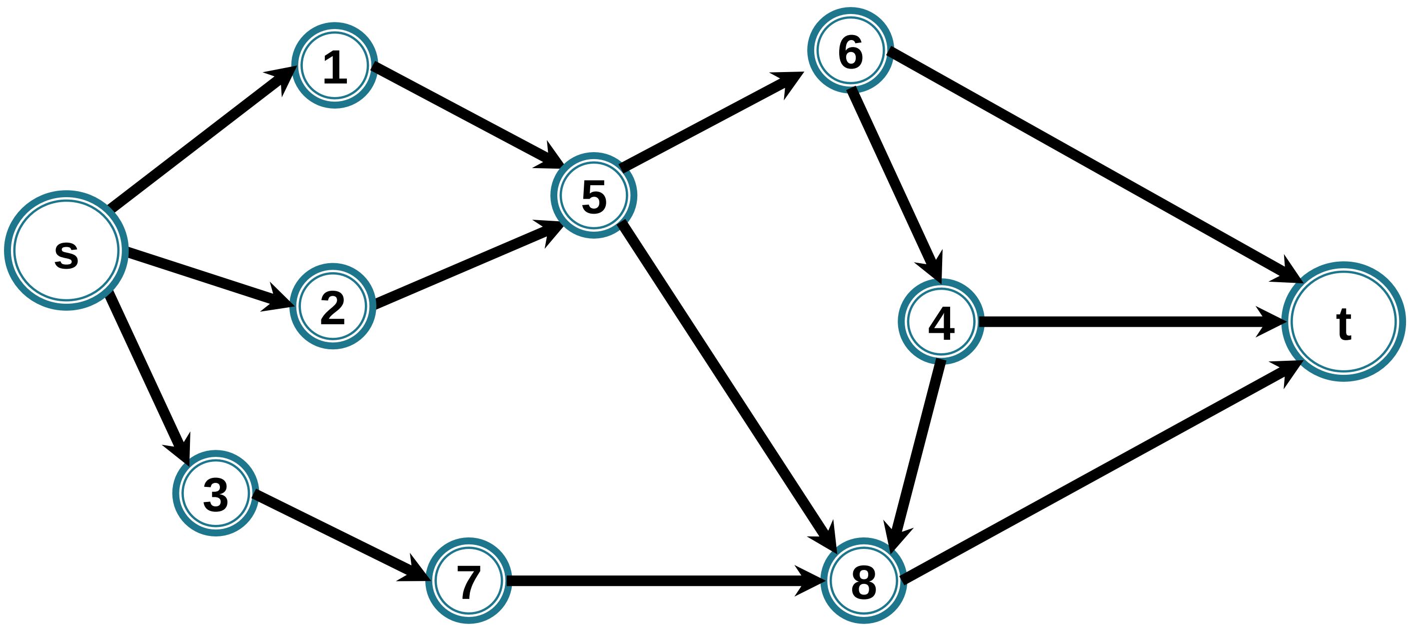
s-t drumuri arc-disjuncte

- $G = (V, E)$ - orientat, conex (graful neorientat suport)
- s, t - două vârfuri

Să se determine numărul maxim k de **s-t drumuri elementare arc-disjuncte (+ k astfel de drumuri)**

- Două drumuri P_1, P_2 s.n. **arc-disjuncte** dacă $E(P_1) \cap E(P_2) = \emptyset$

s-t drumuri arc-disjuncte



s-t drumuri arc-disjuncte

- G= (V, E) – orientat, conex (graful neorientat suport)
- s, t – două vârfuri

Să se determine numărul maxim k de **s-t drumuri elementare arc-disjuncte (+ k astfel de drumuri)**

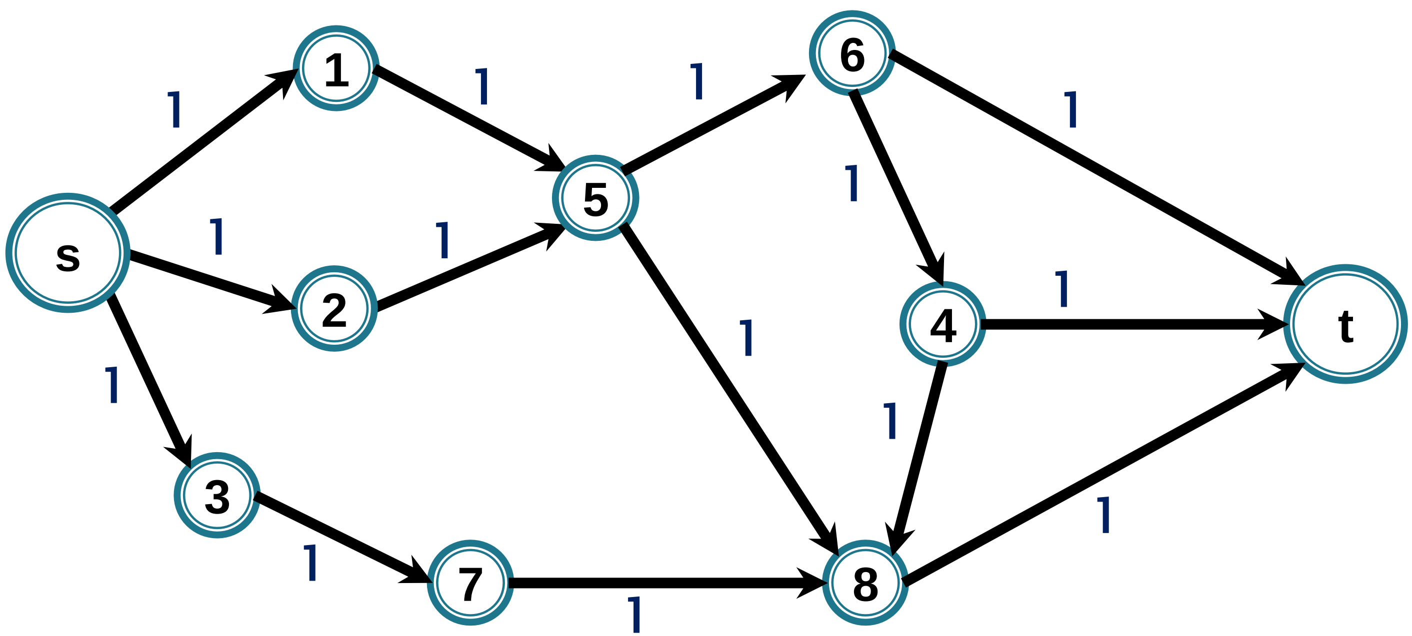
- **Aplicaţii**

- Fiabilitatea reţelelor, conectivitate
- Probleme de strategie
- Măsuri de centralitate (a unui nod) în reţele sociale

    - Jon Kleinberg, Éva Tardos, **Algorithm Design**, Addison-Wesley 2005

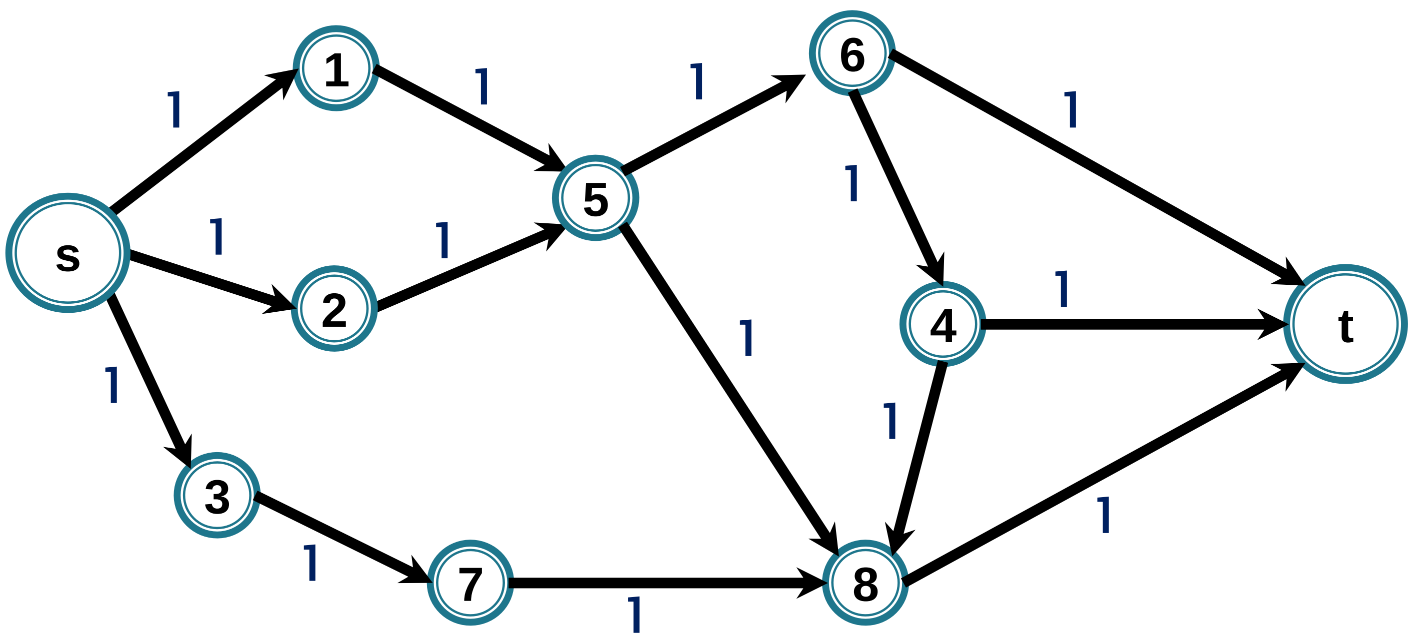
s-t drumuri arc-disjuncte

- **Intuitiv**:
  - Asociem fiecărui arc capacitatea 1
  - Fluxul maxim: f(e) $\in$ {0, 1}

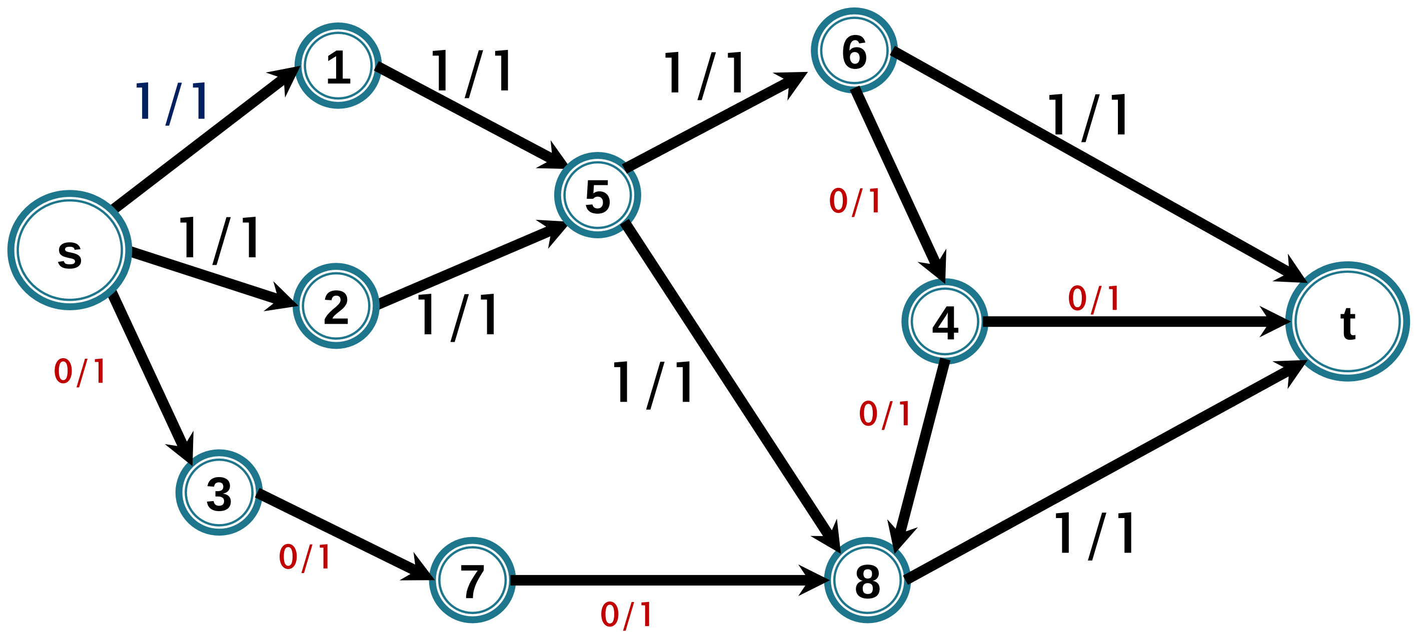


s-t drumuri arc-disjuncte

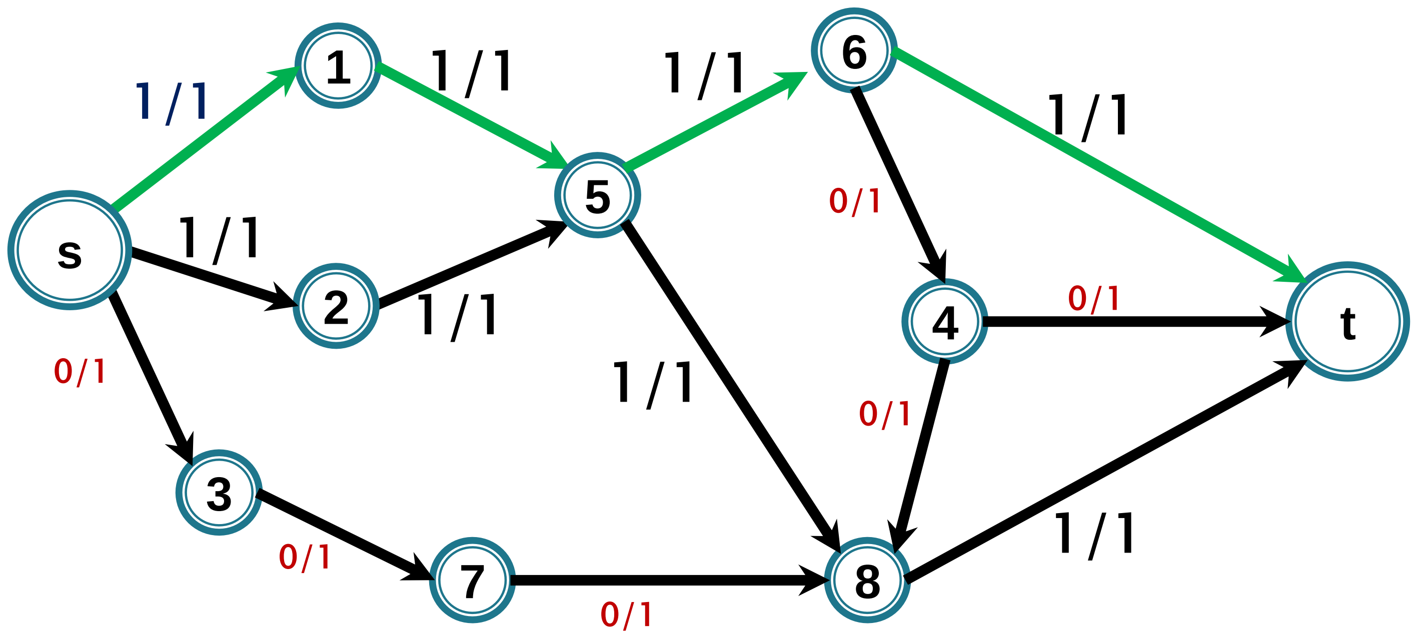
- **Intuitiv**:
  - Asociem fiecărui arc capacitatea 1
  - Fluxul maxim: f(e) $\in$ {0, 1}
  - Un drum de la s la t= **traseul parcurs de o unitate de flux de la s la t**
  - **Numărul de s-t drumuri arc-disjuncte**= valoarea fluxului maxim



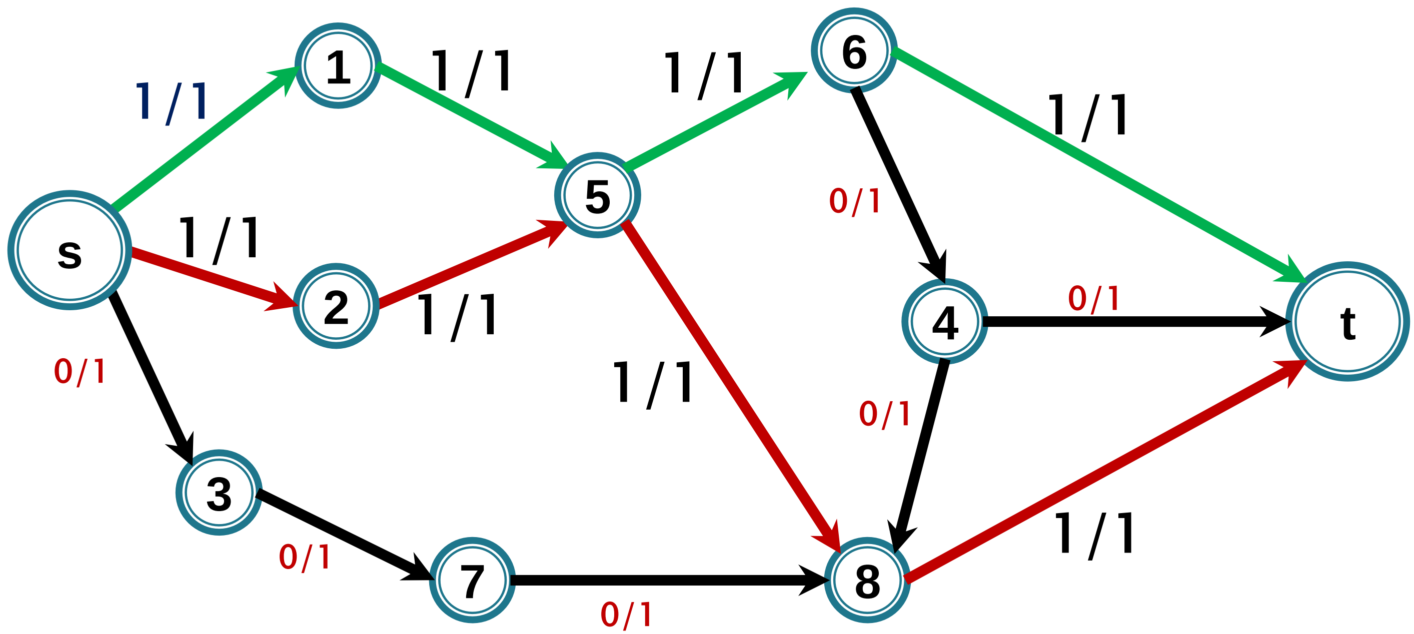
s-t drumuri arc-disjuncte



s-t drumuri arc-disjuncte



s-t drumuri arc-disjuncte



s-t drumuri arc-disjuncte

- **Teorema lui Menger**

  Fie G graf orientat, s, t două vârfuri distincte în G.

  Numărul minim de arce care trebuie eliminate pentru ca s şi t să nu mai fie conectate printr-un drum (să fie **separate**) =

  numărul maxim de drumuri arc-disjuncte de la s la t

s-t drumuri arc-disjuncte

- <u>**Teorema lui Menger**</u>

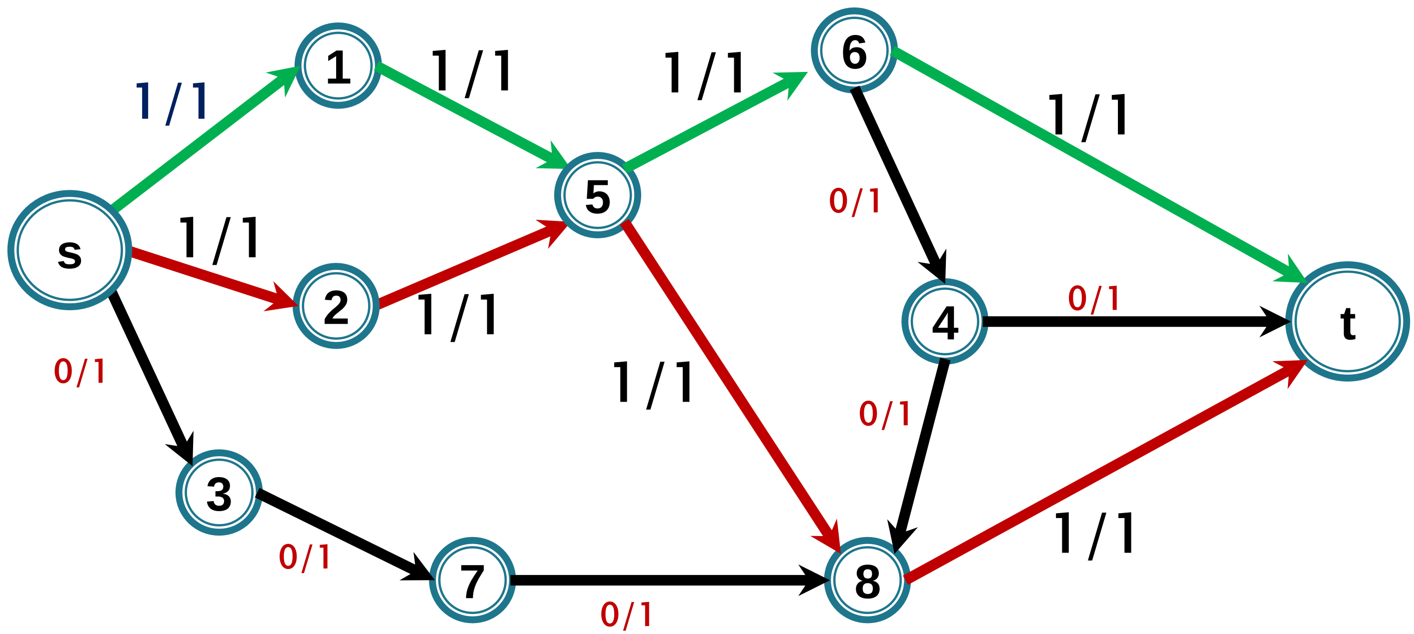
  Fie G graf orientat, s, t două vârfuri distincte în G.

  Numărul minim de arce care trebuie eliminate pentru ca s şi t să nu mai fie conectate printr-un drum (să fie **separate**) =

  numărul maxim de drumuri arc-disjuncte de la s la t

O astfel de mulţime de arce se poate determina cu algoritmul Ford Fulkerson?

s-t drumuri arc-disjuncte



s-t drumuri arc-disjuncte

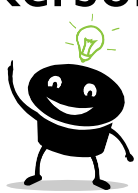
- <u>**Teorema lui Menger**</u>

  Fie G graf orientat, s, t două vârfuri distincte în G.

  Numărul minim de arce care trebuie eliminate pentru ca s şi t să nu mai fie conectate printr-un drum (să fie **separate**) =

  numărul maxim de drumuri arc-disjuncte de la s la t

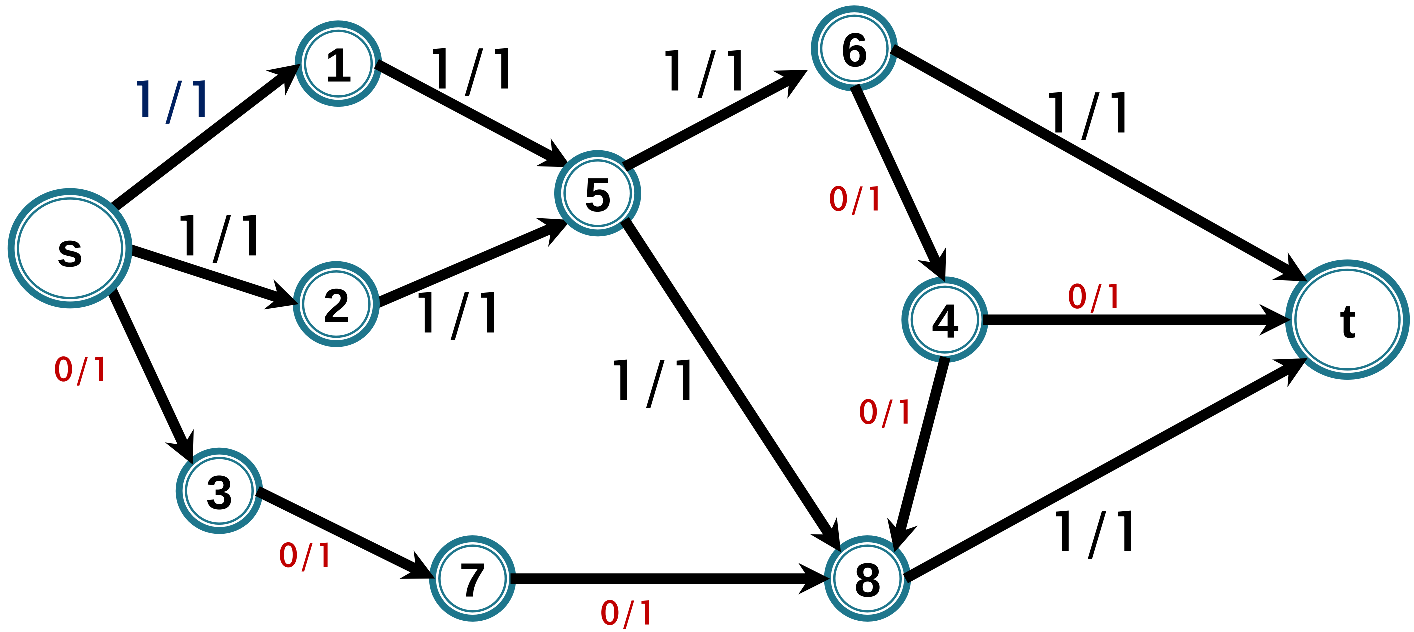
O astfel de mulţime de arce se poate determina cu algoritmul Ford Fulkerson



Sunt arcele directe ale tăieturii minime

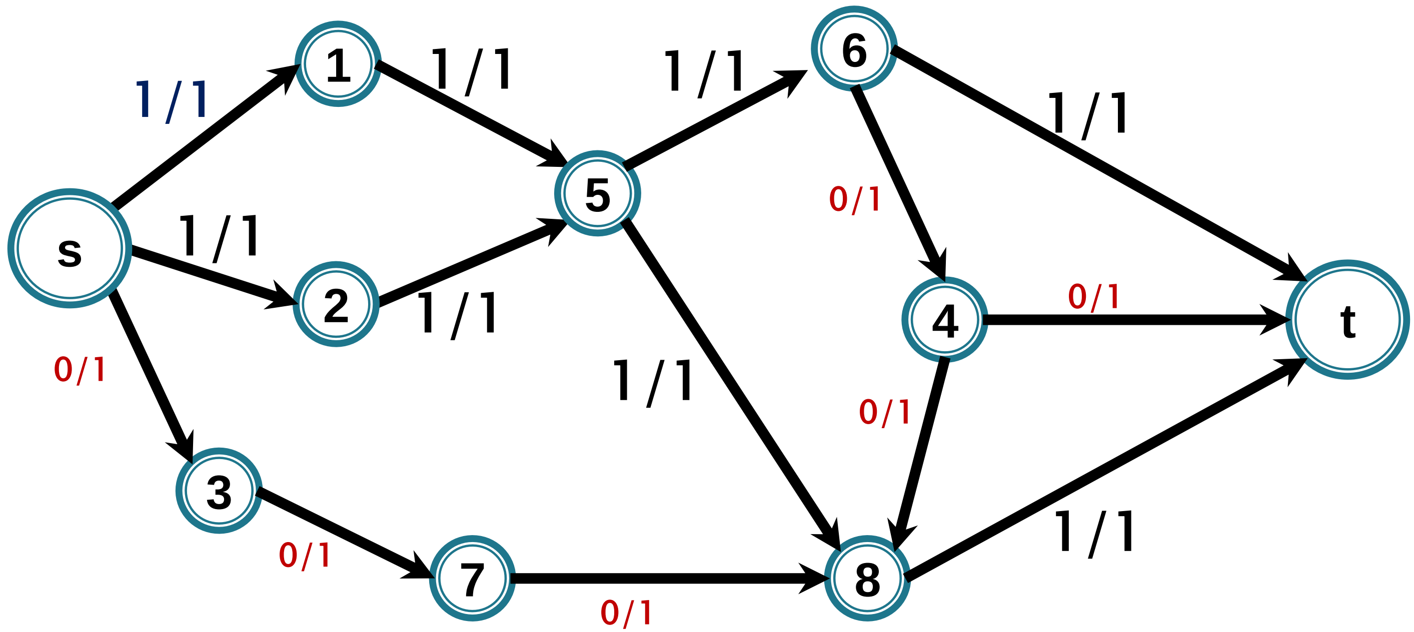
s-t drumuri arc-disjuncte

Cum determinăm tăietura minimă?



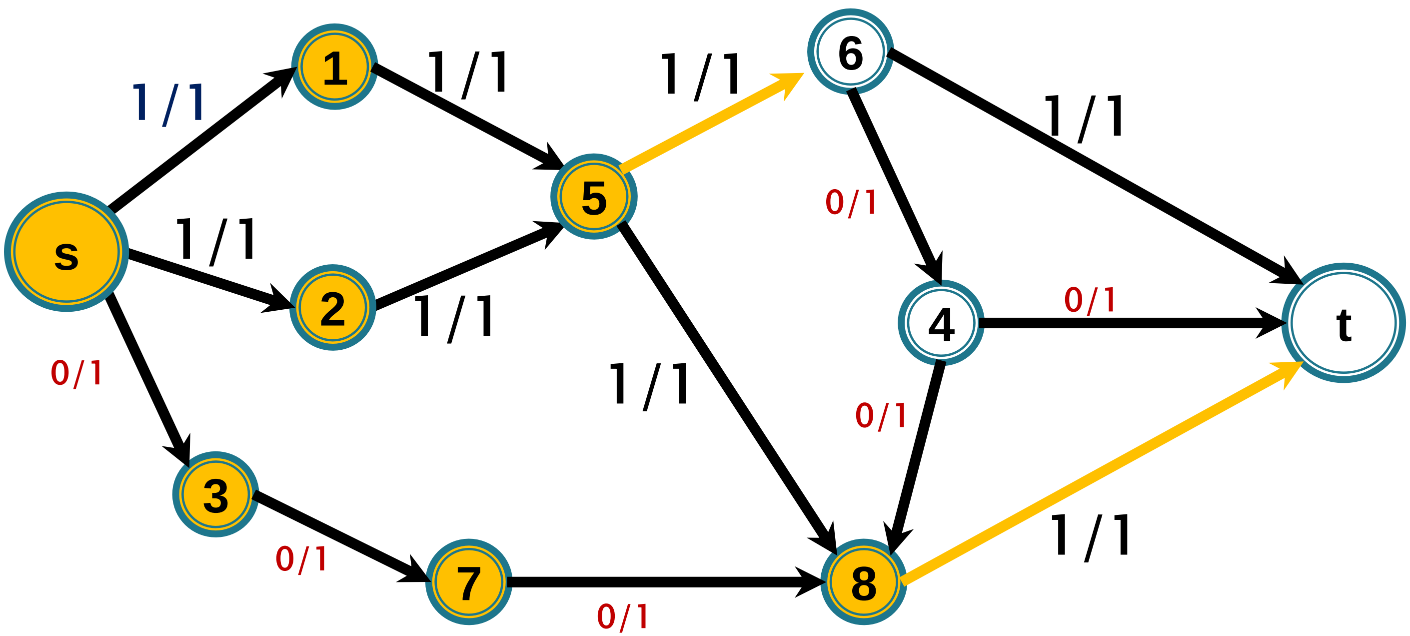
s-t drumuri arc-disjuncte

Mulţimea vârfurilor accesibile din s prin lanţuri f-nesaturate



s-t drumuri arc-disjuncte

Mulţimea vârfurilor accesibile din s prin lanţuri f-nesaturate



s-t drumuri arc-disjuncte

Variante

- Aceeaşi problemă pentru
  - G= (V, E) – **neorientat** conex, |E|>2

- Aceeaşi problemă pentru **vârfuri** (s-t drumuri care nu au vârfuri interne în comun)

s-t drumuri arc-disjuncte

- **Muchie-conectivitatea lui G k'(G) =** cardinalul **minim** al unei mulţimi de muchii $F \subseteq E$ cu proprietatea că

  G – F nu mai este conex

- Dacă k' (G) $\geq$ t, G se numeşte **t-muchie conex**

  - **Amintim (laborator+seminar):**
    - există muchie critică $\Rightarrow$ G este 1-conex
    - Nu există muchie critică $\Rightarrow$ G este 2-conex

- **Cu ajutorul algoritmului de flux maxim putem determina (muchie)-conectivitatea unui graf**