Universitatea Politehnica din București
Facultatea de Automatică și Calculatoare
Departamentul de Calculatoare

ALGORITMUL FORD-FULKERSON

Introducere

- Algoritmul **Ford-Fulkerson** este un algoritm **greedy** care calculează **fluxul maxim** (*maximum flow*) într-o **rețea de transport** (*flow network*)
- A fost descoperit în 1956 de doi matematicieni americani L.R.Ford Jr. și D.R. Fulkerson

Rețea de transport

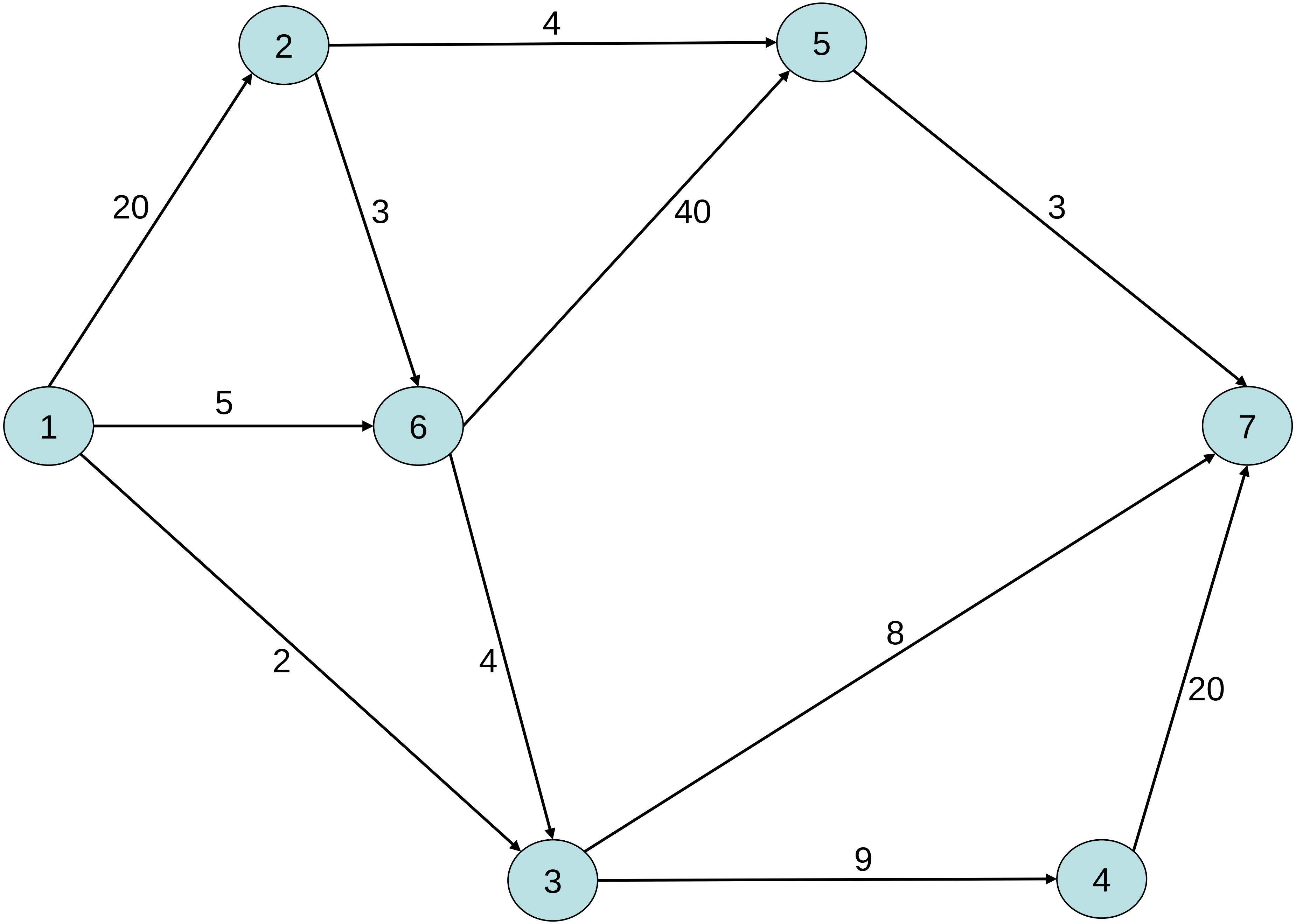
- O **rețea de transport** este un graf orientat `G=(V,E)` care îndeplinește următoarele condiții:
- **1**. Există un singur vârf în graf cu gradul interior `0` (acest vârf este denumit vârf-sursă sau intrare în rețea)

Rețea de transport

- **2**. Există un singur vârf în graf cu gradul exterior 0 (acest vârf este denumit vârf-destinație sau ieșire din rețea)
- **3**. Graful este conex și există drumuri de la vârful-sursă la vârful-destinație
- **4**. Există o funcție $c: E \to \Re^{+}$, prin care se asociază fiecărui arc din graf un număr real pozitiv, denumit **capacitatea** arcului respectiv

Exemplu

- Graful orientat din figură reprezintă o rețea de transport, unde vârful `1` este intrarea rețelei, iar vârful `7` este ieșirea rețelei


1
2
3
4
5
6
7
20
4
3
40
3
5
2
4
8
9
20

Flux

- **Fluxul** într-o rețea de transport este definit de o funcție $f:E \rightarrow \Re^{+}$, care îndeplinește următoarele condiții:
- **1**. **Condiția de mărginire a fluxului**: $f(x,y) \leq c(x,y)$ pentru orice $(x,y) \in E$, adică pentru orice arc din graf valoarea fluxului nu poate depăși capacitatea arcului respectiv

Flux

- **2**. **Condiția de conservare a fluxului**: pentru orice vârf x din graf, exceptând intrarea și ieșirea rețelei, suma fluxurilor arcelor care au ca extremitate inițială vârful x este egală cu suma fluxurilor arcelor care au ca extremitate finală vârful x:
- $\sum_{(x,y)\in E} f(x, y) = \sum_{(y,x)\in E} f(y, x)$

Observație

- Datorită condiției de conservare a fluxului în rețea, suma fluxurilor arcelor care au ca extremitate inițială vârful `s` este egală cu suma fluxurilor arcelor care au ca extremitate finală vârful `d` (această sumă este denumită valoarea fluxului)

Flux maxim

- Se consideră o rețea de transport, reprezentată sub forma unui graf orientat și ponderat, `G=(V,E)`, iar `c` este funcția care reprezintă capacitatea arcelor
- Fluxul de valoare maximă poate fi determinat cu algoritmul **Ford-Fulkerson**

Ideea algoritmului

- Se inițializează fluxul în rețea cu 0
- Cât timp este posibil:
- **1**. Se determină un drum de ameliorare a fluxului
- **2**. Se ameliorează fluxul de-a lungul acestui drum

Drum de ameliorare

- Pentru a determina un drum de ameliorare, J.Edmonds și R.M.Karp au dezvoltat un procedeu de marcaj, bazat pe o parcurgere în lățime a grafului, începând cu intrarea rețelei

Procedeu de marcaj

- Se marchează intrarea rețelei cu $+$
- Fie x un vârf marcat
- Se marchează cu $+x$ orice vârf y nemarcat cu proprietatea că există un arc nesaturat de la x la y, adică $(x,y) \in E$ și $f(x,y) < c(x,y)$

Procedeu de marcaj

- Se marchează cu $-x$ orice vârf y nemarcat cu proprietatea că există un arc cu flux nenul de la y la x, adică $(y,x) \in E$ și $f(y,x) > 0$
- Dacă prin acest procedeu de marcaj s-a marcat ieșirea rețelei, atunci fluxul curent nu este maxim

Observație

- Pe baza etichetelor vârfurilor, se reconstituie un drum `D` de la intrarea rețelei, până la ieșire și se modifică fluxurile arcelor care constituie acest drum

Notații

- $a=\min\{c(x,y)-f(x,y) | y$ marcat cu $+x$, $(x,y)\in D\}$
- $b=\min\{f(y,x) | y$ marcat cu $-x$, $(y,x)\in D\}$
- $v=\min\{a,b\}$
- Din modul de determinare a drumului, $v>0$

Observații

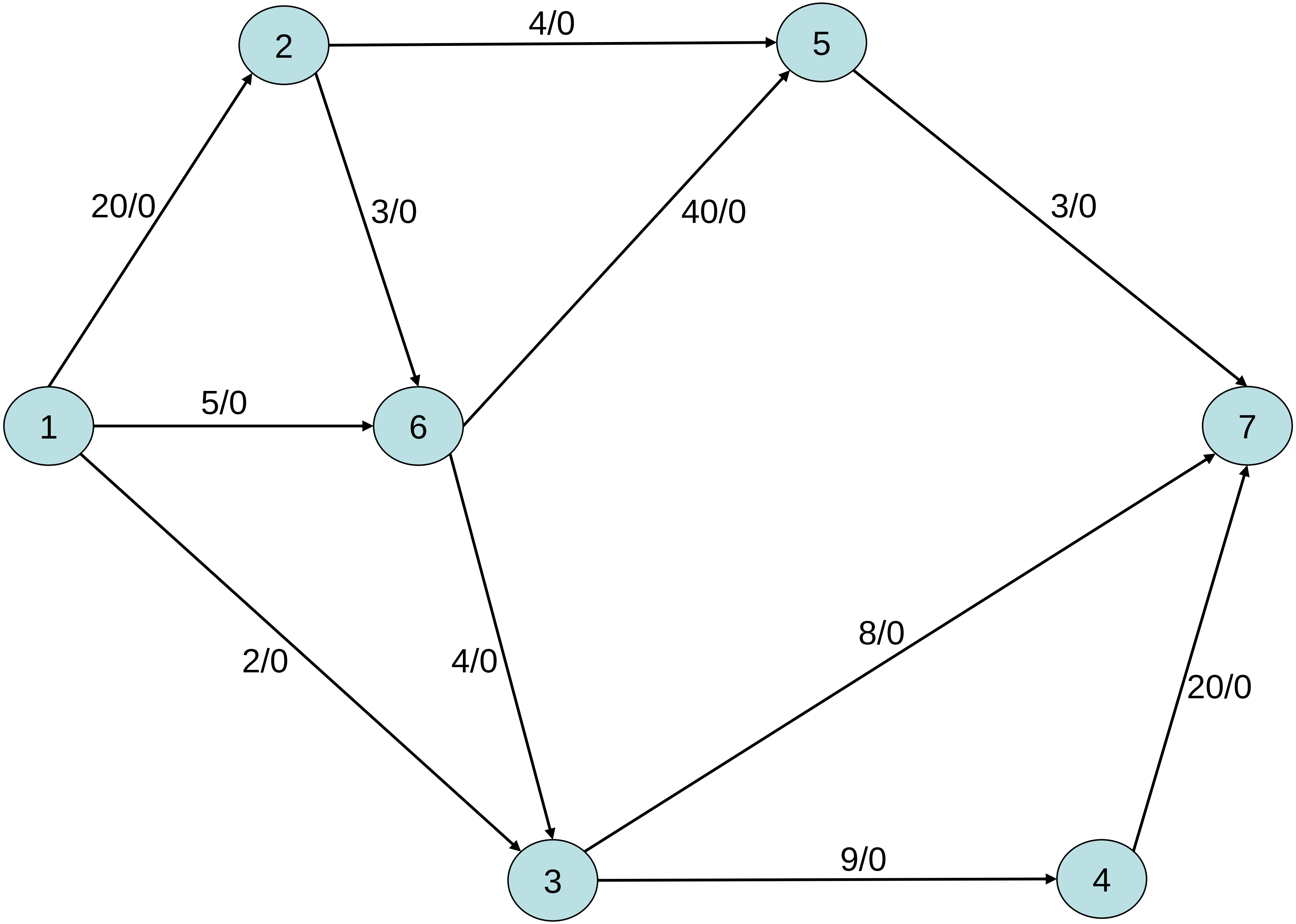
- Se mărește cu `v` fluxul arcelor care aparțin drumului și sunt orientate în sensul sursă→destinație, adică arcele `(x,y)` unde `y` este marcat cu `+x`
- Pentru a respecta condiția de conservare a fluxului, se micșorează cu `v` fluxul arcelor care aparțin drumului și sunt orientate în sensul destinație→sursă, adică arcele `(y,x)` unde `y` este marcat cu `-x`

Observație

- Din modul de definire a noului flux, se observă că respectă condiția de mărginire și de conservare și are valoarea cu v mai mare decât fluxul inițial

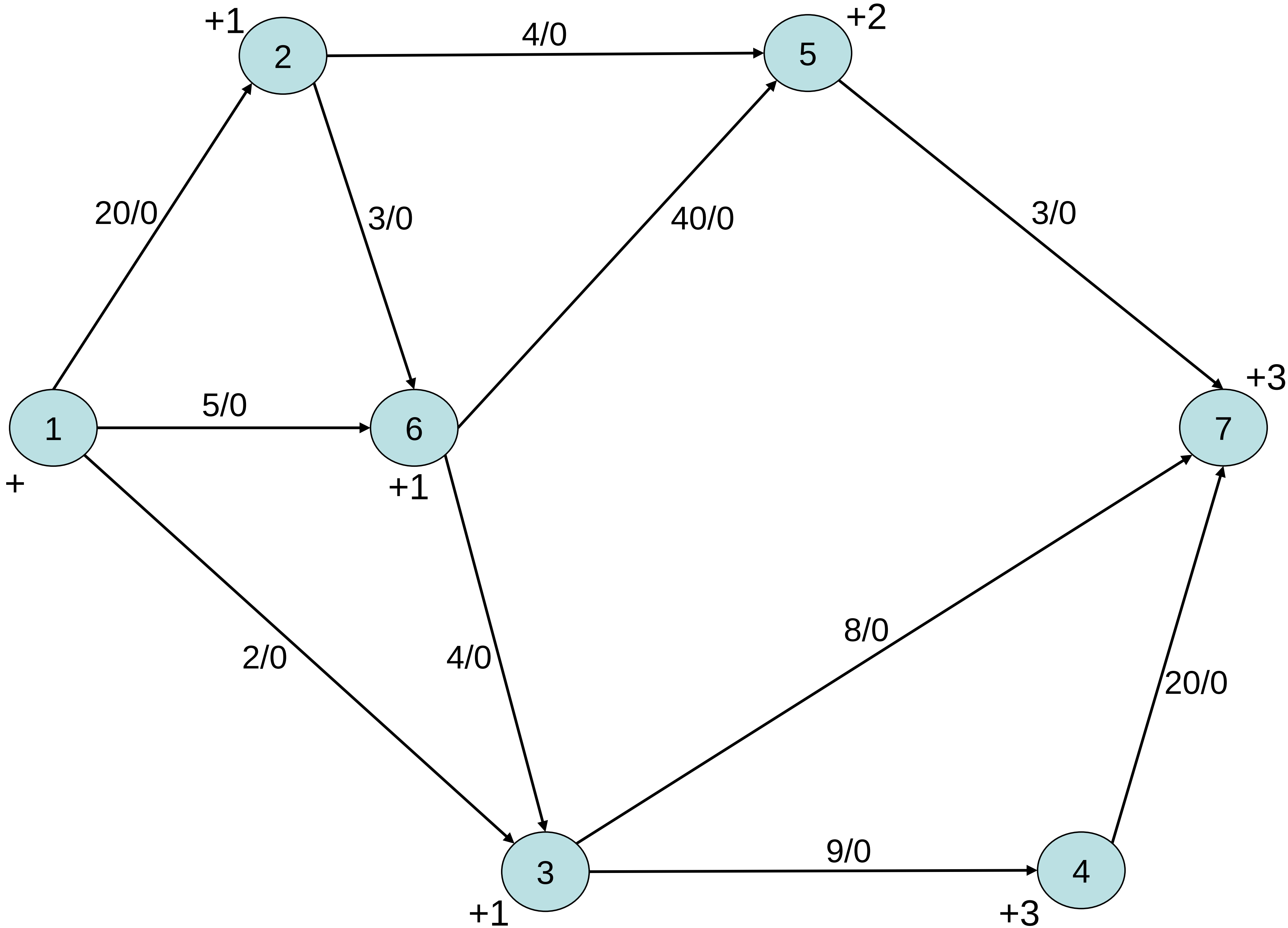
Exemplu

- Se aplică algoritmul pentru rețeaua de transport din exemplul precedent
- Pe fiecare arc este specificată capacitatea, urmată de fluxul arcului respectiv


1
2
3
4
5
6
7
20/0
4/0
3/0
5/0
40/0
3/0
2/0
4/0
8/0
20/0
9/0

Observație

- Printr-o parcurgere în lățime din `1` se marchează vârfurile grafului

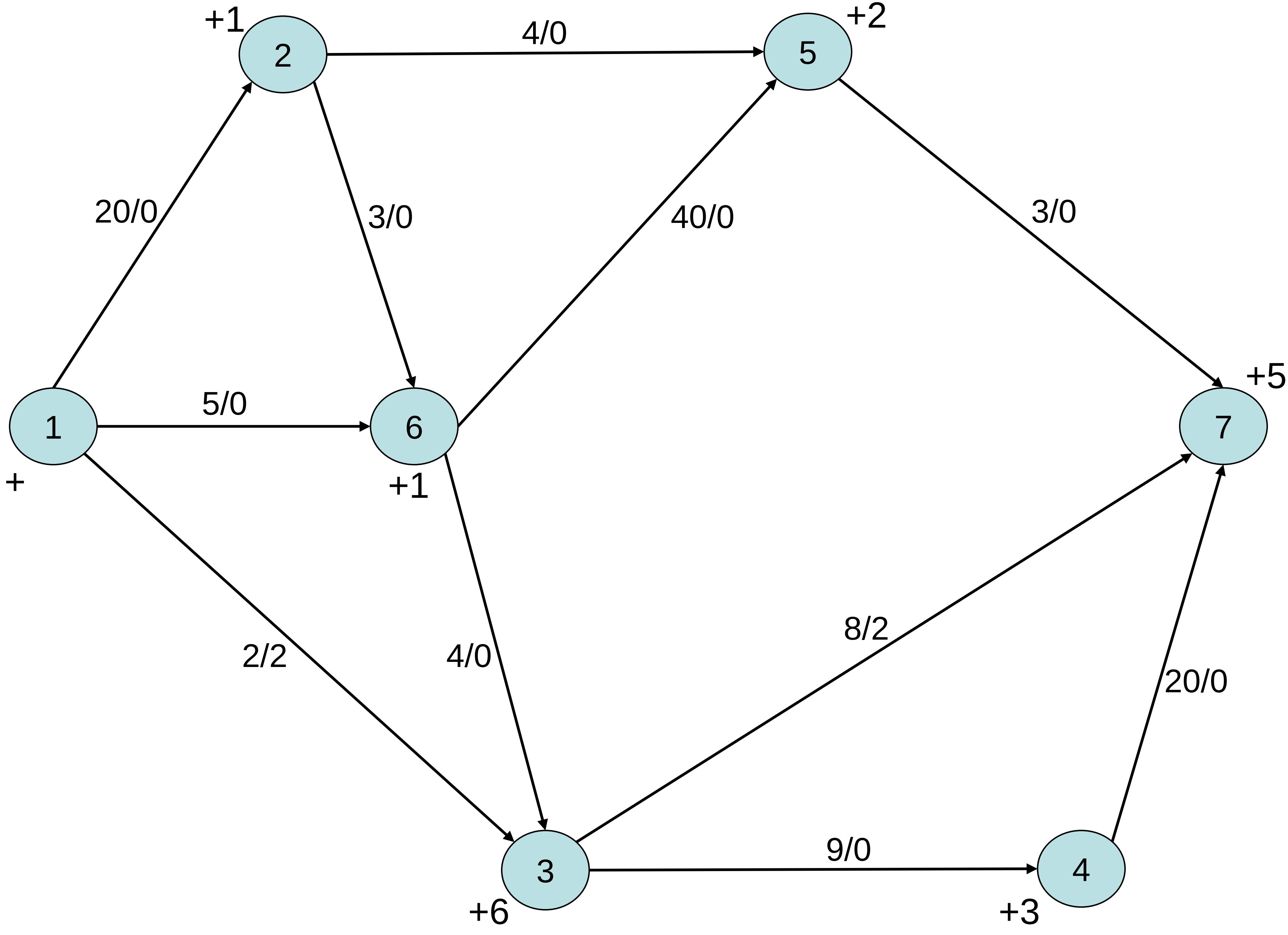

+1
2
4/0
5
+2
20/0
3/0
40/0
3/0
+3
5/0
1
6
7
+
+1
8/0
2/0
4/0
20/0
9/0
3
4
+1
+3

Observații

- S-a marcat ieșirea rețelei (vârful `7`)
- Se reconstituie drumul de ameliorare: `(1,3,7)`
- Toate arcele sunt orientate în sensul sursă$\rightarrow$destinație

Observații

- Se calculează:
- min $\{c(1,3)-f(1,3), c(3,7)-f(3,7)\}$ = min $\{2, 8\}$ = 2
- Se mărește fluxul fiecărui arc de pe acest drum cu `2` și se reia procedeul de marcaj
- Arcul `(1,3)` a devenit saturat (fluxul este egal cu capacitatea)



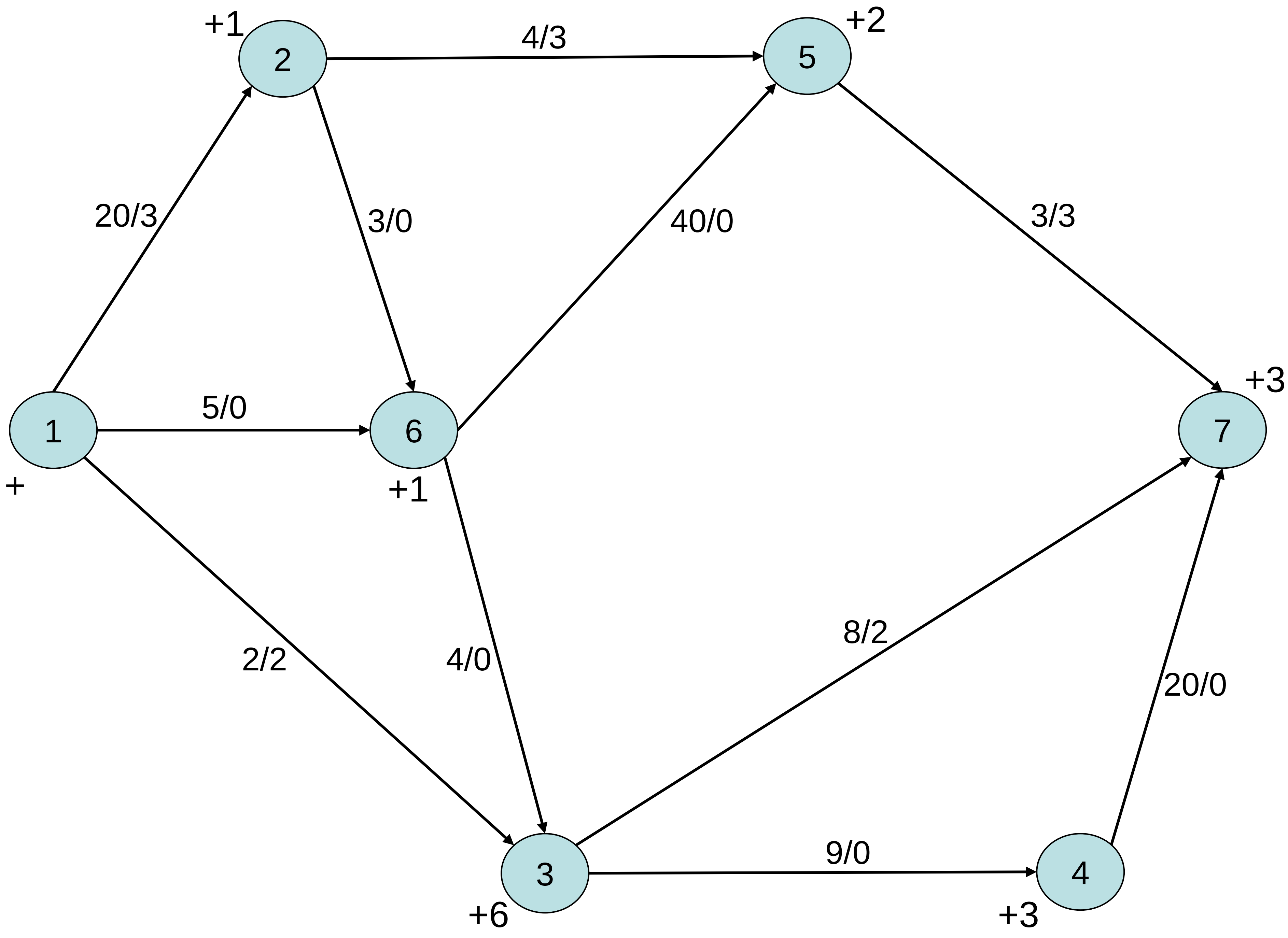
+1
2
4/0
5
+2
20/0
3/0
40/0
3/0
+5
5/0
1
6
7
+
+1
2/2
4/0
8/2
20/0
9/0
3
4
+6
+3

Observații

- Este marcată din nou ieșirea rețelei
- Se reconstituie drumul de ameliorare: `(1,2,5,7)`
- Toate arcele sunt orientate în sensul sursă$\rightarrow$destinație

Observații

- Se calculează:
- min {$c(1,2)-f(1,2)$, $c(2,5)-f(2,5)$, $c(5,7)-f(5,7)$} = min {20, 4, 3} = 3
- Se mărește fluxul fiecărui arc de pe acest drum cu 3 și se reia procedeul de marcaj
- Arcul (5,7) a devenit saturat (fluxul este egal cu capacitatea)

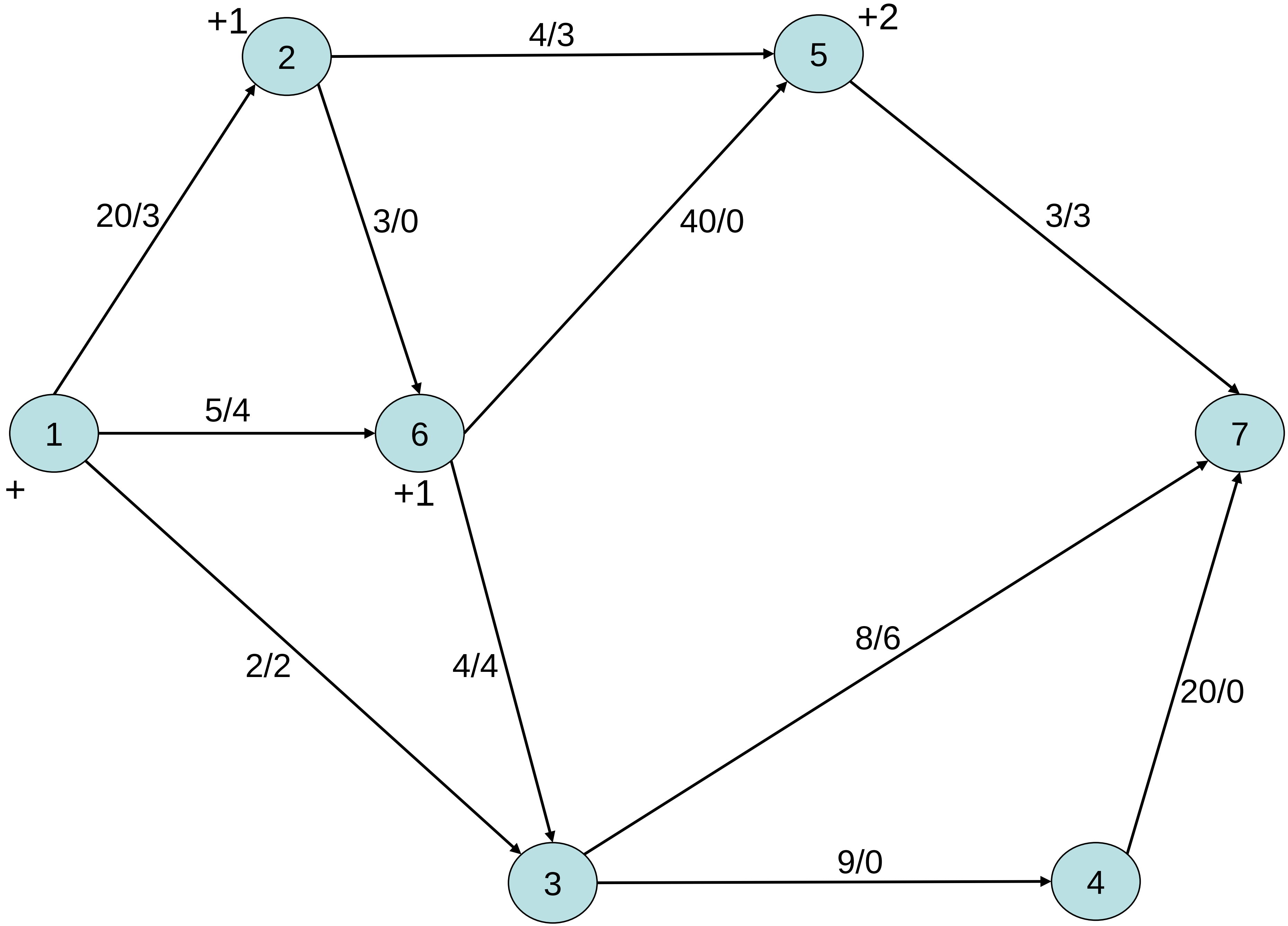

+1
2
4/3
5
+2
20/3
3/0
40/0
3/3
+3
1
5/0
6
7
+
+1
2/2
4/0
8/2
20/0
9/0
3
4
+6
+3

Observații

- Este marcată din nou ieșirea rețelei
- Se reconstituie drumul de ameliorare: `(1,6,3,7)`
- Toate arcele sunt orientate în sensul sursă$\rightarrow$destinație

Observații

- Se calculează:
- min $\{c(1,6)-f(1,6), c(6,3)-f(6,3), c(3,7)-f(3,7)\} = \min\{5, 4, 6\} = 4$
- Se mărește fluxul fiecărui arc de pe acest drum cu 4 și se reia procedeul de marcaj
- Arcul (6,3) a devenit saturat (fluxul este egal cu capacitatea)


+1
2
4/3
5
+2
20/3
3/0
40/0
3/3
5/4
1
6
7
+
+1
2/2
4/4
8/6
20/0
9/0
3
4

Observații

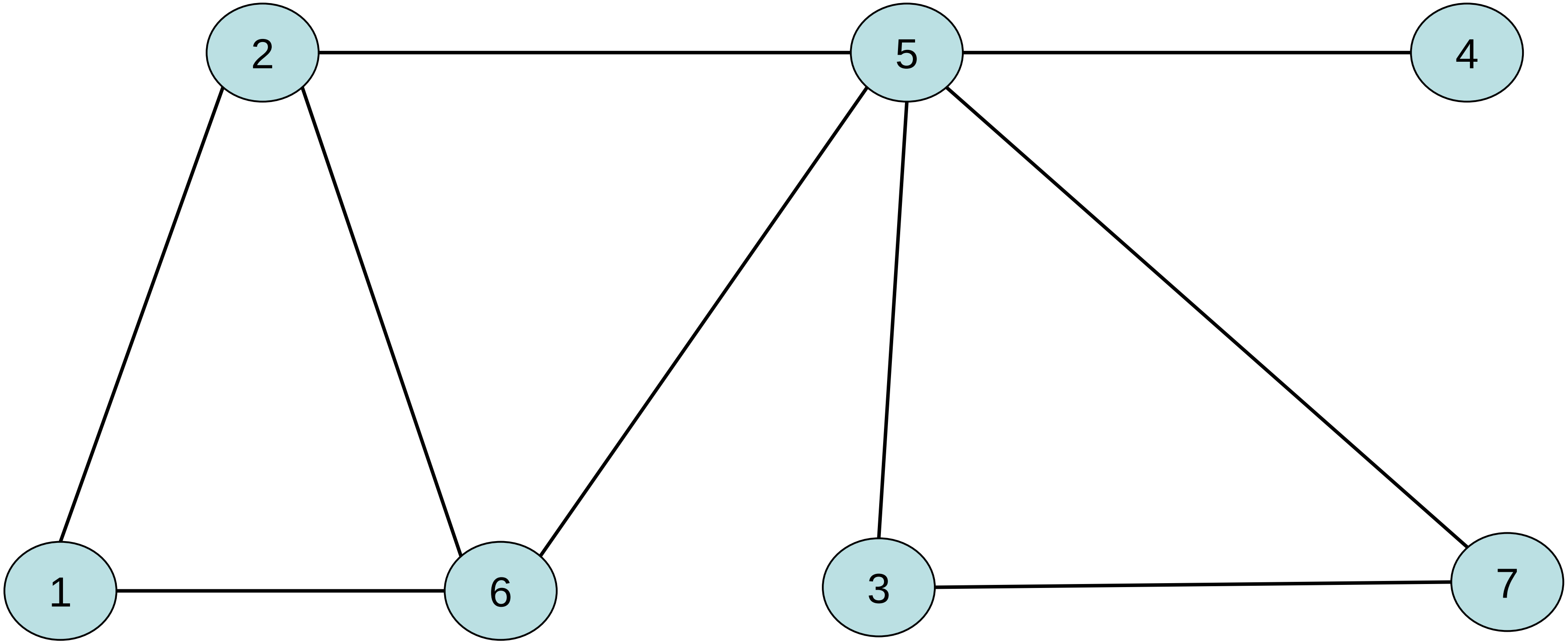
- La această nouă parcurgere nu se poate marca vârful `7`
- `9` este fluxul maxim în rețea
- Complexitatea algoritmului este $O(n*m^2)$

Cuplaj maximal în graf bipartit

- Se consideră `G` un graf neorientat
- Se numește **cuplaj** în graful `G` o submulțime a muchiilor sale cu proprietatea că nu există două muchii care să aibă o extremitate comună
- Cuplajul se numește **maximal** dacă numărul de muchii din submulțime este maxim

Exemplu

- Mulțimea {`(2,6),(3,5)`} este un cuplaj
- Un cuplaj maximal este {`(1,6),(3,7),(4,5)`}



Observații

- Pentru cazul în care graful este bipartit, adică mulțimea vârfurilor poate fi partiționată în două mulțimi `A`, `B`, astfel încât orice muchie din graf are o extremitate în mulțimea `A` și cealaltă extremitate în mulțimea `B`, problema găsirii cuplajului maximal poate fi redusă la determinarea fluxului maxim într-o rețea de transport

Observații

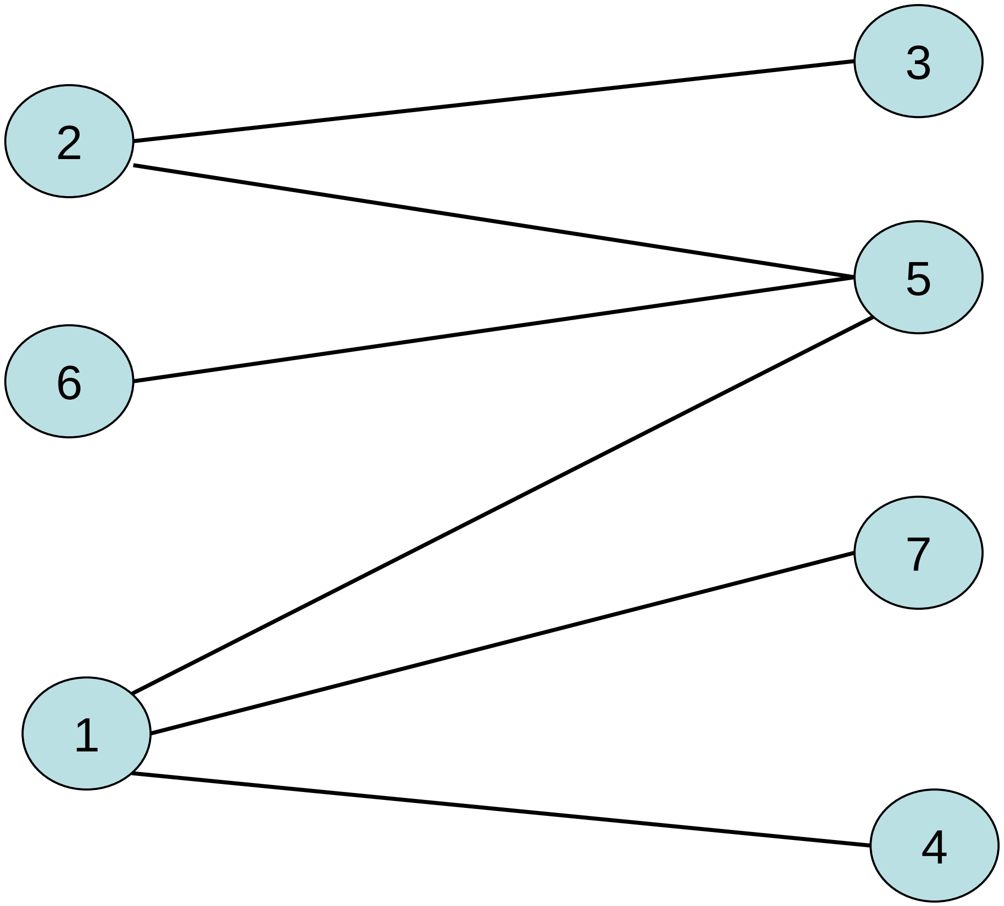
- Se construiește o rețea de transport în care:
- **1**. Mulțimea vârfurilor este mulțimea vârfurilor grafului `G`, la care adăugăm două vârfuri fictive `s` și `d` (sursa și destinația)

Observații

- **2**. Mulțimea arcelor grafului este formată din:
  - Arcele care au sursa drept extremitate inițială și vârfurile din mulțimea `A` ca extremitate finală
  - Arcele care au ca extremitate finală destinația și ca extremitate inițială fiecare vârf din mulțimea `B`
  - Arcele care corespund muchiilor grafului `G`, considerând ca extremitate inițială vârful din mulțimea `A` și ca extremitate finală vârful din mulțimea `B`
- **3**. Fiecare arc va avea capacitatea `1`

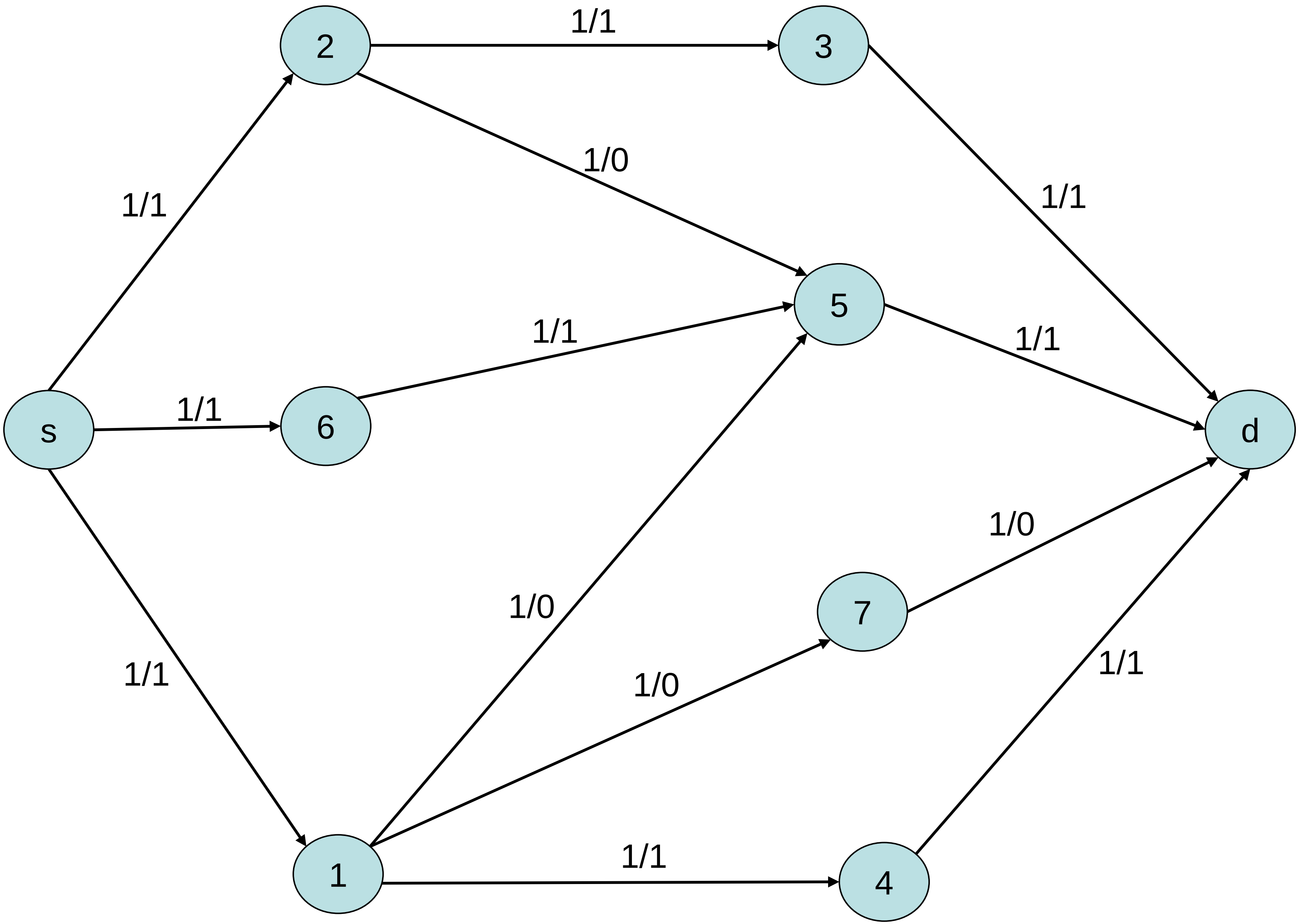
Exemplu

- Se consideră un graf bipartit



Exemplu

- Rețeaua de transport asociată acestui graf bipartit este completată cu un flux maximal în rețea
- Prin determinarea unui flux maxim în această rețea de transport, se determină practic un cuplaj în graf, format din toate muchiile corespunzătoare arcelor de la mulțimea `A` la mulțimea `B`, având flux nenul



s
2
3
5
6
d
7
1
4
1/1
1/0
1/1
1/1
1/1
1/1
1/1
1/0
1/0
1/0
1/1
1/1
1/1