



1

## Fluxuri în rețele

- Drumuri de creștere
- Secțiuni
- Teorema drumului de creștere
- Teorema fluxului întreg
- Teorema flux maxim – secțiune minimă
- Algoritmul lui Ford & Fulkerson
- Algoritmul lui Edmonds & Karp

2

## Exerciții pentru seminarul 9 (săptămâna 2 - 5 decembrie)



## Definiție

Fie  $R = (G, s, t, c)$  o rețea și  $x$  un flux în  $R$ . Un **A-drum** (în  $R$  relativ la fluxul  $x$ ) este un drum  $P$  în  $\widetilde{G}$  astfel încât  $\forall ij \in E(P)$ :

- dacă  $ij$  este un arc direct, atunci  $x_{ij} < c_{ij}$ ,
- dacă  $ij$  este a arc invers, atunci  $x_{ji} > 0$ .

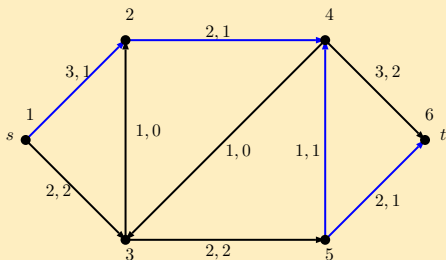
Dacă  $P$  este un A-drum în  $R$  relativ la fluxul  $x$ , atunci **capacitatea reziduală** a arcului  $ij \in E(P)$  este

$$r(ij) = \begin{cases} c_{ij} - x_{ij}, & \text{dacă } ij \text{ este un arc direct al lui } P \\ x_{ji}, & \text{dacă } ij \text{ este un arc invers al lui } P \end{cases}$$

**Capacitatea reziduală** a drumului  $P$  este  $r(P) = \min_{e \in E(P)} r(e)$ .

## Exemplu

Considerăm rețeaua de mai jos, unde, pe fiecare arc, prima etichetă este capacitatea iar a doua este fluxul.



$P : 1, 12, 2, 24, 4, 45, 5, 56, 6$  este un  $A$ -drum de la  $s$  la  $t$  cu arcele forward  $12$  ( $x_{12} = 1 < c_{12} = 3$ ),  $24$  ( $x_{24} = 1 < c_{24} = 2$ ),  $56$  ( $x_{56} = 1 < c_{56} = 2$ ), și arcul backward  $45$  ( $x_{45} = 1 > 0$ ). Capacitatea reziduală a lui  $P$ :  $r(P) = \min \{2, 1, 1, 1\} = 1$ .

## Definiție

Un drum de creștere relativ la fluxul  $x$  în rețeaua  $R = (G, s, t, c)$  este un  $A$ -drum de la  $s$  la  $t$ .

## Lema 1

Dacă  $P$  este un drum de creștere relativ la fluxul  $x$  în  $R = (G, s, t, c)$ , atunci  $x^1 = x \otimes r(P)$  definit prin

$$x_{ij}^1 = \begin{cases} x_{ij}, & \text{dacă } ij \notin E(P) \\ x_{ij} + r(P), & \text{dacă } ij \in E(P), ij \text{ este arc direct în } P \\ x_{ij} - r(P), & \text{dacă } ij \in E(P), ij \text{ este arc invers în } P \end{cases}$$

este un flux în  $R$  cu  $v(x^1) = v(x) + r(P)$ .

**Demonstrație.** Din definiția lui  $r(P)$ , constrângerile (i) sunt satisfăcute de  $x^1$ . Constrângerile (ii) - verificate de  $x$  - nu sunt afectate pentru  $x^1$  în vreun nod  $i \notin V(P)$ .

Pentru  $i \in V(P) \setminus \{s, t\}$  există exact două arce în  $P$  incidente cu  $i$ , e. g.  $li$  și  $ik$ . Avem următoarele două cazuri posibile:

a)  $li$  și  $ik$  sunt arce directe:

$$\begin{aligned} & \sum_j x_{ji}^1 - \sum_j x_{ij}^1 = \sum_{j \neq l} x_{ji} - \sum_{j \neq k} x_{ij} + x_{li}^1 - x_{ik}^1 \\ & = \sum_{j \neq l} x_{ji} - \sum_{j \neq k} x_{ij} + x_{li} + r(P) - x_{ik} - r(P) = \sum_j x_{ji} - \sum_j x_{ij} = 0. \end{aligned}$$

b)  $li$  este arc direct și  $ik$  arc invers:

$$\begin{aligned} & \sum_j x_{ji}^1 - \sum_j x_{ij}^1 = \sum_{j \neq l, k} x_{ji} - \sum_j x_{ij} + x_{li}^1 + x_{ki}^1 \\ & = \sum_{j \neq l, k} x_{ji} - \sum_j x_{ij} + x_{li} + r(P) + x_{ki} - r(P) = \sum_j x_{ji} - \sum_j x_{ij} = 0. \end{aligned}$$

## Problema fluxului maxim - Drumuri de creștere

c)  $li$  arc invers și  $ik$  arc direct: similar cu b).

d)  $li$  și  $ik$  arce inverse: similar cu a).

$v(x^1)$  diferă de  $v(x)$  datorită fluxului de pe arcul  $lt$  din  $P$ :

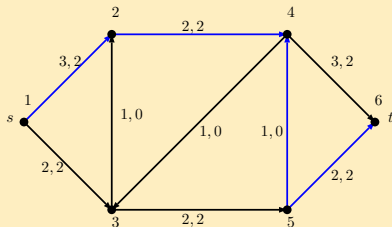
$lt$  arc direct:

$$\begin{aligned}v(x^1) &= \sum_j x_{jt}^1 - \sum_j x_{tj}^1 = \sum_{j \neq l} x_{jt} - \sum_j x_{tj} + x_{lt}^1 = \\ &= \sum_{j \neq l} x_{jt} - \sum_j x_{tj} + x_{lt} + r(P) = v(x) + r(P).\end{aligned}$$

$lt$  arc invers:

$$\begin{aligned}v(x^1) &= \sum_j x_{jt}^1 - \sum_j x_{tj}^1 = \sum_j x_{jt} - \sum_{j \neq l} x_{tj} - x_{tl}^1 = \\ &= \sum_j x_{jt} - \sum_{j \neq l} x_{tj} - (x_{tl} - r(P)) = v(x) + r(P). \quad \square\end{aligned}$$

Pentru exemplul de mai sus, fluxul  $x^1 = x \otimes r(P)$  de valoare  $v(x^1) = v(x) + r(P) = 3 + 1 = 4$  este:



\* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph

## Remarci

- Lema de mai sus explică și numele drumurilor de creștere și capacitatea reziduală.
- Din definiție, dacă  $P$  este un drum de creștere, atunci  $r(P) > 0$  și  $v(x \otimes r(P)) > v(x)$ . Urmează că **dacă există un drum de creștere relativ la fluxul  $x$ , atunci  $x$  nu este un flux de valoare maximă.**

### Definiție

Fie  $R = (G, s, t, c)$  o rețea. O **secțiune** în  $R$  este o partiție  $(S, T)$  a lui  $V(G)$  cu  $s \in S$  și  $t \in T$ . **Capacitatea secțiunii**  $(S, T)$  este

$$c(S, T) = \sum_{i \in S} \sum_{j \in T} c_{ij}.$$

### Lema 2

Dacă  $x$  este un flux în  $R = (G, s, t, c)$  și  $(S, T)$  este o secțiune în această rețea, atunci

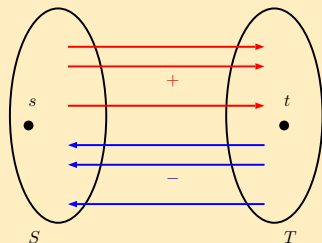
$$v(x) = \sum_{i \in S} \sum_{j \in T} (x_{ij} - x_{ji})$$

(valoarea fluxului este fluxul net care trece prin secțiune).

# Problema fluxului maxim - Secțiuni

C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms

\* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms



Demonstrație.

$$\begin{aligned} v(x) &= \left( \sum_j x_{sj} - \sum_j x_{js} \right) + 0 = \\ &= \left( \sum_j x_{sj} - \sum_j x_{js} \right) + \sum_{i \in S, i \neq s} \left( \sum_j x_{ij} - \sum_j x_{ji} \right) = \end{aligned}$$



**Demonstrație.** Din Lema 2

$$\begin{aligned}
 v(x) &= \sum_{i \in S} \sum_{j \in T} (x_{ij} - x_{ji}) \stackrel{x_{ij} \leq c_{ij}}{\leq} \sum_{i \in S} \sum_{j \in T} (c_{ij} - x_{ji}) \stackrel{x_{ji} \geq 0}{\leq} \\
 &\leq \sum_{i \in S} \sum_{j \in T} c_{ij} = c(S, T). \quad \square
 \end{aligned}$$

**Remarcă**

Dacă  $\bar{x}$  este un flux în  $R$  și  $(\bar{S}, \bar{T})$  este o secțiune astfel încât  $v(\bar{x}) = c(\bar{S}, \bar{T})$ , atunci,  $\forall x$  flux în  $R$ , avem  $v(x) \leq c(\bar{S}, \bar{T}) = v(\bar{x})$ , i. e.,  $\bar{x}$  este un flux de valoare maximă în  $R$ .

Similar,  $\forall (S, T)$  secțiune în  $R$ , avem  $c(S, T) \geq v(\bar{x}) = c(\bar{S}, \bar{T})$ , adică,  $(\bar{S}, \bar{T})$  este o secțiune de capacitate minimă în  $R$ .

## Teorema 1

Un flux  $x$  este un flux de valoare maximă dacă și numai dacă nu există un drum de creștere relativ la  $x$  în  $R$ .

**Demonstrație.** " $\Rightarrow$ " Dacă  $P$  este un drum de creștere relativ la  $x$ , atunci  $x \otimes r(P)$  este un flux în  $R$  de valoare strict mai mare.

" $\Leftarrow$ " Fie  $x$  un flux în  $R$  cu proprietatea că nu există drum de creștere relativ la  $x$  în  $R$ . Fie

$$S = \{i : i \in V \text{ și } \exists P \text{ un } A\text{-drum în } R \text{ de la } s \text{ la } i\}.$$

Evident  $s \in S$  (există un  $A$ -drum de lungime 0 de la  $s$  la  $s$ ) și  $t \notin S$  (nu există vreun  $A$ -drum de la  $s$  la  $t$ ). Astfel, luând  $T = V \setminus S$ ,  $(S, T)$  este o secțiune în  $R$ .



## Problema fluxului maxim - Teorema fluxului întreg

C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms  
\* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph

### Teorema 2

(Teorema fluxului întreg) Dacă toate capacitățile din  $R$  sunt întregi atunci există un flux întreg,  $x$ , de valoare maximă (toate  $x_{ij} \in \mathbb{Z}_+$ ).

Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru -

Demonstrație. Considerăm următorul algoritm:

$x^0 \leftarrow 0; i \leftarrow 0;$

while ( $\exists P_i$  un drum de creștere relativ la  $x^i$ ) do

$x^{i+1} \leftarrow x^i \otimes r(P_i); i ++;$

end while

Să observăm că " $x^i$  are doar componente întregi" este un invariant al algoritmului (din definiția lui  $r(P_i)$ , dacă toate capacitățile sunt întregi și  $x^i$  este întreg, atunci  $r(P_i)$  este întreg și astfel  $x^{i+1}$  este întreg).

- Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \*



## Teorema 3

(Teorema flux maxim – secțiune minimă) Valoarea maximă a unui flux în rețeaua  $R = (G, s, t, c)$  este egală cu capacitatea minimă a unei secțiuni în  $R$ .

**Linia demonstrației.** Dacă descriem un algoritm care, plecând cu un flux inițial  $x^0$  (e.g.,  $x^0 = 0$ ), construiește într-un număr finit de pași un flux  $x$  fără drumuri de creștere, atunci secțiunea considerată în demonstrația Teoremei 1 satisface, împreună cu  $x$ , cerința teoremei.

Pentru capacități raționale, algoritmul considerat în demonstrația Teoremei 2 satisface această condiție. Pentru capacități arbitrare reale vom prezenta mai târziu un astfel de algoritm datorat lui **Edmonds și Karp (1972)**.

C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms  
\* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph  
Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru -

### Remarci

- O demonstrație scurtă a teoremei de mai sus constă în a arăta că există un flux de valoare maximă și de a aplica construcția din demonstrația Teoremei 1. Un flux de valoare maximă există întotdeauna observând că acesta este soluția optimă a unei probleme de programare liniară (peste un politop nevid).
- Importanța algoritmică a Teoremei 3 este dată de faptul că mulțimea tuturor secțiunilor dintr-o rețea este finită, pe când mulțimea tuturor fluxurilor este infinită.

C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms  
C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru -  
Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru  
- Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \*

## Problema fluxului maxim - Algoritmul lui Ford & Fulkerson

C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms  
\* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph  
Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph

Algoritmul întreține o etichetare a nodurilor din rețea pentru a determina drumuri de creștere relativ la fluxul curent  $x$ . Când nu mai există drumuri de creștere, fluxul  $x$  este de valoare maximă.

Fie  $R = (G = (V, E), s, t, c)$  o rețea și  $x$  un flux în  $R$ .

Eticheta unui nod  $j$ , care are trei componente  $(e_1, e_2, e_3)$ , semnifică: există un  $A$ -drum de la  $s$  la  $j$ ,  $P$ , unde  $e_1 = i$  este nodul dinaintea lui  $j$  pe acest drum,  $e_2 \in \{direct, invers\}$  reprezintă direcția arcului  $ij$ , iar  $e_3 = r(P)$ .

Inițial  $s$  este etichetat  $(\cdot, \cdot, \infty)$ . Celelalte noduri primesc eventual etichete prin vizitarea (scanarea) nodurilor deja etichetate:

Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \*

```
procedure scan(i)
for (j ∈ V, neetichetat) do
  if (ij ∈ E și  $x_{ij} < c_{ij}$ ) then
    etichetează j cu  $e = (i, \text{direct}, \min \{e_3[i], c_{ij} - x_{ij}\})$ ;
  end if
  if (ji ∈ E și  $x_{ji} > 0$ ) then
    etichetează j cu  $e = (i, \text{invers}, \min \{e_3[i], x_{ji}\})$ ;
  end if
end for
```

Semnificația componentelor etichetelor este întreținută de procedura **scan**.

Când, în procedura **scan**, nodul *t* este etichetat, se poate detecta un drum de creștere *P*, relativ la fluxul curent *x*, astfel:



pornește cu un flux inițial  $x = (x_{ij})$  (e. g.,  $x = 0$ )

$e(s) \leftarrow (\cdot, \cdot, \infty)$ ;

while ( $\exists$  noduri etichetate și nescanate ) do

**alege**  $i$  un nod etichetat și nescanat;

    scan( $i$ );

    if ( $t$  a fost etichetat) then

        modifică fluxul pe drum dat de etichete;

        șterge toate etichetele;  $e(s) \leftarrow (\cdot, \cdot, \infty)$ ;

    end if

end while

$S \leftarrow \{i : i \in V, i \text{ este etichetat}\}$ ;

$T \leftarrow V \setminus S$ ;

$x$  este un flux de valoare maximă,  $(S, T)$  este o secțiune de capacitate minimă.

## Problema fluxului maxim - Algoritmul lui Ford & Fulkerson

C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms

**Complexitatea timp:** Fiecare creștere a fluxului curent necesită cel mult  $2m$  ( $m = |E|$ ) inspecții ale arcelor pentru etichetarea altor noduri. Dacă toate capacitățile sunt întregi, sunt necesare cel mult  $v$  creșteri ( $v$  fiind valoarea fluxului maxim). Astfel algoritmul are complexitatea timp  $\mathcal{O}(mv)$ .

Dacă  $U$  este un majorant al tuturor capacităților pe arce, atunci  $v \leq (n - 1)U$  (acesta este un majorant al secțiunii  $(\{s\}, V \setminus \{s\})$ ), deci algoritmul are complexitatea timp  $\mathcal{O}(nmU)$ .

Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru -

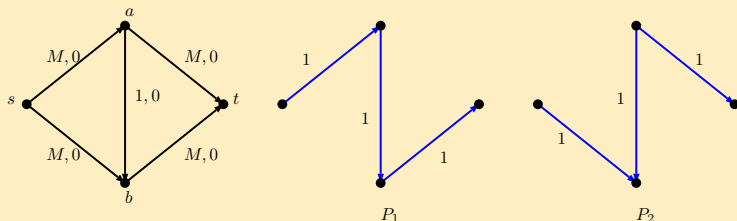
### Remarcă

Este posibil ca algoritmul să nu se termine pentru capacități iraționale. Această situație nu apare în implementările practice, dar neajunsul acestei descrieri a algoritmului este dat de faptul că numărul de creșteri ale fluxului curent depinde de capacități (și nu de dimensiunile rețelei).

# Problema fluxului maxim - Algoritmul lui Ford & Fulkerson

C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms  
\* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph  
Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru -

## Exemplu



Dacă operația **alege** din algoritmul de mai sus determină drumurile de creștere  $P_1, P_2, P_1, P_2, \dots$ , unde  $P_1 = (s, sa, a, ab, b, bt, t)$  și  $P_2 = (s, sb, b, ba, a, at, t)$ , atunci fiecare creștere a fluxului se face cu 1, și algoritmul necesită  $2M$  creșteri, ceea ce este prea mult.

Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru  
- Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \*

C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms

\* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph

Aceste deficiențe ale algoritmului pot fi evitate dacă alegerea nodurilor etichetate în vederea scanării nu se face aleatoriu (Dinic (1970), Edmonds & Karp (1972)).

\* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph

## Definiție

Un cel mai scurt drum de creștere relativ la fluxul  $x$  în  $R$  este un drum de creștere de lungime minimă printre toate drumurile de creștere relativ la  $x$ .

Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru -

Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru -

Fie  $x$  un flux în rețeaua  $R$ . Fie  $x^k$  ( $k \geq 1$ ) secvența de fluxuri:

$$x^1 \leftarrow x;$$

$$x^{k+1} \leftarrow x^k \otimes r(P_k), \quad P_k \text{ un cel mai scurt drum de creștere relativ la}$$

$$x^k;$$

Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms



## Teorema 4

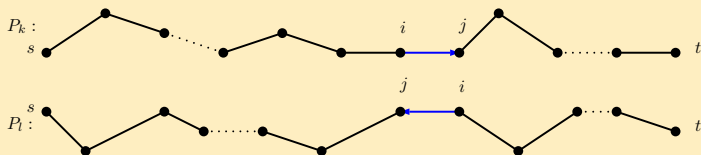
(Edmonds & Karp, 1972) Dacă  $x = x^1$  este un flux arbitrar în rețeaua  $R$ , atunci secvența de fluxuri  $x^1, x^2, \dots$ , obținută din  $x^1$  prin creșteri succesive cu drumuri de creștere de lungime minimă, are cel mult  $mn/2$  termeni (în cel mult  $mn/2$  creșteri succesive se obține un flux  $x$  cu proprietatea că nu există drum de creștere relativ la  $x$ ).

**Demonstrație.** Dacă  $P$  este un drum de creștere relativ la un flux  $x$  în  $R$ , cu capacitatea reziduală  $r(P)$ , un **arc critic** în  $P$  este un arc  $e \in E(P)$  cu  $r(e) = r(P)$ . În  $x \otimes r(P)$ , fluxul de pe arce critice devine fie egal cu capacitatea (pe arcele directe), sau nul (pe arcele inverse).

Fie  $ij$  un arc critic de pe un cel mai scurt drum de creștere  $P_k$  relativ la  $x^k$ . Lungimea lui  $P_k$  este  $\sigma_i^k + \tau_i^k = \sigma_j^k + \tau_j^k$

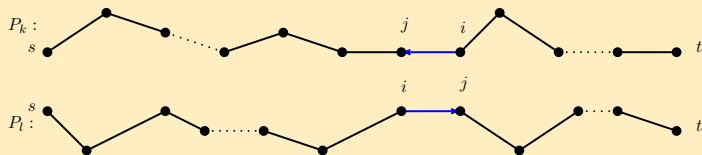
Deoarece  $ij$  este critic în  $P_k$ , în  $x^{k+1}$  nu mai poate fi utilizat în aceeași direcție ca în  $P_k$ . Fie  $P_l$  (cu  $l > k$ ) primul cel mai scurt drum de creștere relativ la  $x^l$  în care fluxul de pe arcul  $ij$  va fi modificat, (când arcul va fi folosit în direcție inversă decât în  $P_k$ ). Avem două cazuri:

$ij$  este un arc direct în  $P_k$ . Atunci  $\sigma_j^k = \sigma_i^k + 1$ ; în  $P_l$   $ij$  va fi arc invers, deci  $\sigma_i^l = \sigma_j^l + 1$ .



Urmează că  $\sigma_i^l + \tau_i^l = \sigma_j^l + 1 + \tau_i^l \geq \sigma_j^k + 1 + \tau_i^k = \sigma_i^k + \tau_i^k + 2$  (din Lema 4). Am obținut că  $length(P_l) \geq length(P_k) + 2$ .

$ij$  este un arc invers în  $P_k$ . Atunci  $\sigma_i^k = \sigma_j^k + 1$ ; în  $P_l$   $ij$  va fi arc direct, deci  $\sigma_j^l = \sigma_i^l + 1$ .



Urmează că  $\sigma_j^l + \tau_j^l = \sigma_i^l + 1 + \tau_j^l \geq \sigma_i^k + 1 + \tau_j^k = \sigma_j^k + \tau_j^k + 2$ . Obținem că  $length(P_l) \geq length(P_k) + 2$ .

Astfel orice cel mai scurt drum de creștere pe care arcul  $ij$  este critic are lungimea cu cel puțin 2 mai mare decât lungimea drumului precedent pe care  $ij$  a fost critic. Deoarece lungimea unui drum în  $G$  este cel mult  $n - 1$ , urmează că un arc fixat nu poate fi critic de mai mult de  $n/2$  ori (de-a lungul întregului proces de creștere).

Orice drum de creștere are cel puțin un arc critic. Astfel în secvența  $(P_k)$  avem cel mult  $mn/2$  drumuri de creștere cele mai scurte.

## Corolar

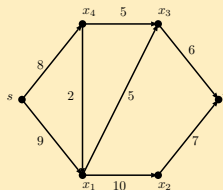
*În orice rețea există un flux  $x$  cu proprietatea că nu există drumuri de creștere relativ la  $x$ .*

## Remarci

- Demonstrația Teoremei 4 este acum încheiată.
- Singura modificare din **Algoritmul lui Ford & Fulkerson** este în alegerea nodului etichetat care va fi scanat: se folosește regula "primul etichetat-primul scanat" adică, se întreține o coadă a nodurilor etichetate (inițializată cu  $s$ , la fiecare început de creștere).



**Exercițiul 1.** Considerăm rețeaua de transport de mai jos:



Determinați un flux maxim în rețeaua de mai jos utilizând algoritmul lui Edmonds-Karp pentru următoarele ordonări ale listelor de adiacență:

- (a)  $A(s) = (x_1, x_4)$ ,  $A(x_1) = (x_3, x_2, x_4)$ ,  $A(x_2) = (x_1, t)$ ,  
 $A(x_3) = (x_1, x_4, t)$ ,  $A(x_4) = (x_3, x_1)$ .
- (b)  $A(s) = (x_4, x_1)$ ,  $A(x_1) = (x_2, x_3, x_4)$ ,  $A(x_2) = (x_1, t)$ ,  
 $A(x_3) = (x_1, x_4, t)$ ,  $A(x_4) = (x_3, x_1)$ .



### Exercițiul 2 (continuare).

(c) Considerăm următorul algoritm:

```
SC-MAX-FLOW( $R$ ) {  
     $C \leftarrow \max_{e \in E} c(e)$ ;  
     $x \leftarrow 0$ ; //  $x$  fluxul curent;  
     $K \leftarrow 2^{1 + \lceil \log C \rceil}$ ;  
    while( $K \geq 1$ ) {  
        while( $x$  are un drum de creștere  $P$  cu  $r(P) \geq K$ )  
             $x \leftarrow x \otimes r(P)$ ;  
         $K \leftarrow K/2$ ;  
    }  
    return  $x$ ;  
}
```

### Exercițiul 2 (continuare).

1. Arătați că procedura SC-MAX-FLOW( $R$ ) returnează un flux de valoare maximă  $x$  în  $R$ .
2. Arătați că, după fiecare iterație **while** exterioară, valoarea maximă a unui flux în  $R$  este cel mult  $v(x) + m \cdot K$ .
3. Arătați că,  $\forall K \in \mathbb{Z}_+$ , există cel mult  $\mathcal{O}(m)$  iterații **while** interioare. În consecință procedura are complexitatea timp  $\mathcal{O}(m^2 \log C)$ .

**Exercițiul 3.** Digraful  $G = (V, E)$  reprezintă topologia unei rețele de procesoare. Fiecărui procesor,  $v \in V$ , îi cunoaștem încărcarea,  $load(v) \in \mathbb{R}_+$ . Folosind un flux maxim într-o anumită rețea determinați o strategie statică de echilibrare a încărcării (static load balancing strategy) în  $G$ : indicați pentru fiecare procesor încărcarea ce trebuie trimisă și cărui procesor așa încât toate procesoarele să aibă aceeași încărcare.

## Exerciții pentru seminarul 9

C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms  
\* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph

**Exercițiul 4.** Fie  $R = (G, s, t, c)$  o rețea și  $(S_i, T_i)(i = \overline{1, 2})$  două secțiuni de capacitate minimă în  $R$ . Arătați că  $(S_1 \cup S_2, T_1 \cap T_2)$  și  $(S_1 \cap S_2, T_1 \cup T_2)$  sunt de asemenea secțiuni de capacitate minimă.

\* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph  
Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru -

**Exercițiul 5.** Fie  $R = (G = (V, E), s, t, c)$  o rețea și  $c : E \rightarrow \mathbb{Z}_+$ ,  $n = |V|$ ,  $m = |E|$ .

- Descrieți un algoritm de complexitate timp  $\mathcal{O}(n + m)$  pentru a determina o secțiune de capacitate minimă în  $R$ , având la îndemână un flux de valoare maximă  $x^*$ .
- Folosind un algoritm de flux maxim pentru o anumită funcție de capacitate, arătați că se poate găsi o secțiune de capacitate minimă în  $R$ ,  $(S_0, T_0)$ , cu număr minim de arce.

- Graph Algorithms \* C. Croitoru - Graph Algorithms \* C. Croitoru - Graph Algorithms \*



**Exercițiul 8.** Fie  $R = (G, s, t, c)$  o rețea astfel ca  $st, ts \notin E(G)$ . Avem și o funcție majorant  $m : E(G) \rightarrow \mathbb{R}_+$ ,  $m(e) \leq c(e), \forall e \in E(G)$ . Un flux  $\varphi$  în  $R$  se numește flux legal în  $R$  dacă  $x(e) \geq m(e), \forall e \in E(G)$ .

(a) Arătați că, pentru orice flux legal  $\varphi$  și orice  $st$ -secțiune  $(S, T)$  avem

$$v(\varphi) \leq \sum_{i \in S, j \in T, ij \in E(G)} c(ij) - \sum_{i \in S, j \in T, ji \in E(G)} m(ji)$$

(b) Pornind de la  $R$  construim o rețea  $\bar{R} = (\bar{G}, \bar{s}, \bar{t}, \bar{c})$ , unde

- $V(\bar{G}) = V(G) \cup \{\bar{s}, \bar{t}\}$  (acestea sunt noduri noi);
- $E(\bar{G}) = E(G) \cup \{st, ts\} \cup \{\bar{s}v, v\bar{t} : v \in V(G)\}$ ;
- pentru orice  $v \in V(G)$ ,  $\bar{c}(\bar{s}v) = \sum_{uv \in E(G)} m(uv)$  și  $\bar{c}(v\bar{t}) =$

$$\sum_{vu \in E(G)} m(vu);$$

- $\bar{c}(st) = \bar{c}(ts) = \infty$  și pentru orice  $ij \in E(G)$ ,  $\bar{c}(ij) = c(ij) - m(ij)$ .

### Exercițiul 8 (continuare).

Arătați că există un flux legal în  $R$  dacă și numai dacă există un flux în  $\bar{R}$  de valoare  $M = \sum_{e \in E(G)} m(e)$ .

- (c) Folosind un flux legal inițial (vezi (b)) arătați cum se poate modifica algoritmul Ford-Fulkerson pentru a obține un flux legal de valoare maximă.

### Exercițiul 9. Fie $R = (G, s, t, c)$ o rețea de transport.

- (a) Descrieți un algoritm eficient care să verifice dacă o secțiune dată,  $(S, T)$ , este de capacitate minimă în  $R$ .
- (b) Descrieți un algoritm de complexitate timp  $\mathcal{O}(n + m)$  care să verifice dacă un flux dat,  $x$ , este de valoare maximă în  $R$ .
- (c) Fie  $c : E \rightarrow \mathbb{Z}_+$  și  $x^*$  este un flux întreg de valoare maximă în  $R$ . Să presupunem că pe un arc,  $e_0 \in E(G)$ , capacitatea crește cu 1. Descrieți un algoritm eficient care să determine un flux maxim în noua rețea.

