UNIUNEA EUROPEANĂ

GUVERNUL ROMÂNIEI

Instrumente Structurale
2007-2013

Platformă de e-learning și curriculă e-content pentru învățământul superior tehnic

Proiectarea Algoritmilor

24. Fluxuri maxime. Metoda Ford-Fulkerson

Bibliografie

[1] C. Giumale – Introducere in Analiza Algoritmilor - cap. 5.6

[2] Cormen – Introducere in algoritmi - cap. 27

[3] Wikipedia - http://en.wikipedia.org/wiki/Ford-Fulkerson_algorithm

Calculul fluxului maxim

- Metoda Ford-Fulkerson
  - $f(u,v) = 0 \ \forall u,v$
  - **Repetă** // creștere iterativă a fluxului
    - găsește un drum s..p..t pe care se poate mări fluxul (cale reziduală)
      - f = f + flux(s..p..t)
    - **Până când** nu se mai poate găsi nici un drum s..p..t
  - **Întoarce** f

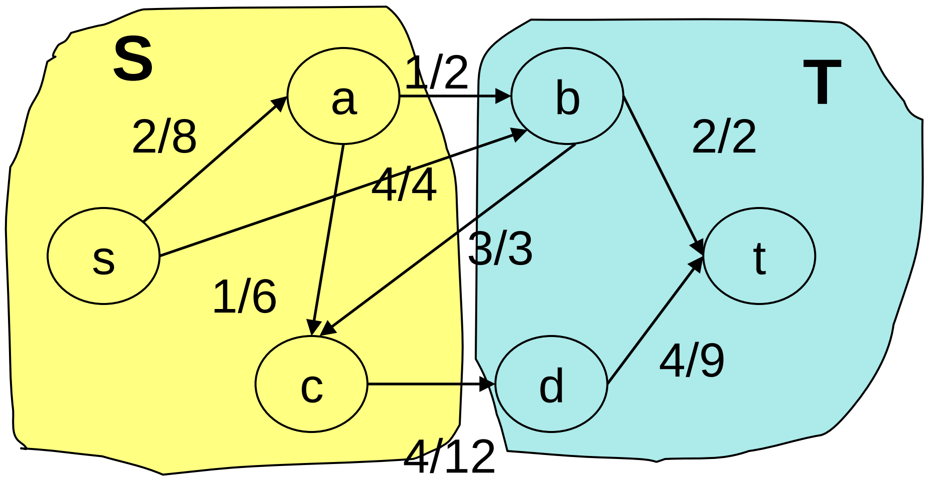
- În funcție de metodele de identificare a căii există mai mulți algoritmi ce urmează această metodă.

PA

Tăieturi in rețele de flux

- Definiție: O tăietură (S,T) a unei rețele de flux G = partiționare a nodurilor în 2 mulțimi disjuncte S și T = V \ S a.î. $s \in S$ și $t \in T$.
  - $f(S,T) = \Sigma_{x \in S} \Sigma_{y \in T} f(x,y)$ – fluxul prin tăietura
  - $c(S,T) = \Sigma_{x \in S} \Sigma_{y \in T} c(x,y)$ – capacitatea tăieturii
- Lema 5.18: Fluxul prin tăietură = fluxul prin rețea – $f(S,T) = |f|$
- Corolar 5.5: S, T – tăietură oarecare – fluxul maxim este limitat superior de capacitatea tăieturii $|f| \le c(S,T)$

Exemplu de tăietură într-o rețea de flux



- f(S,T) = 6 = f(s,V) = f(a,b) + f(s,b) + f(c,d) + f(c,b) = 4 + 1 + 4 - 3 = 6
- c(S,T) = c(a,b) + c(s,b) + c(c,d) = 18

Flux maxim – tăietură minimă

- Teorema 5.25 (Flux maxim – tăietură minimă): G = (V,E) rețea de flux – următoarele afirmații sunt echivalente:
  - f este o funcție de flux în G a.î. |f| este flux maxim total în G;
  - rețeaua reziduală G_f nu are căi reziduale;
  - există o tăietură (S,T) a.î. |f| = c(S,T).

Algoritmul Ford – Fulkerson

- Ford – Fulkerson(G,s,t)
  - **Pentru fiecare** (u,v) din E
    - f(u,v) = f(v,u) = 0 // inițializare
  - **Cât timp**
    - Există o cale reziduală p între s..t în G_f
      - $c_f(p) = \min\{c_f(u,v) \mid (u,v) \text{ din } p\}$ // capacitatea reziduală
      - **Pentru fiecare** (u,v) din p
        - $f(u,v) = f(u,v) + c_f(p)$
        - $f(v,u) = -f(u,v)$
  - **Întoarce** |f|

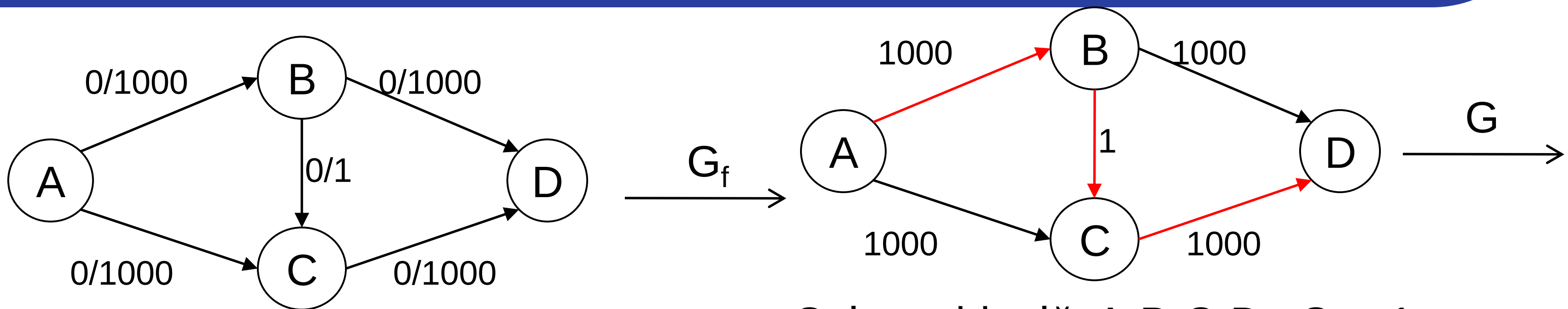
Complexitate?

Algoritmul Ford – Fulkerson (2)

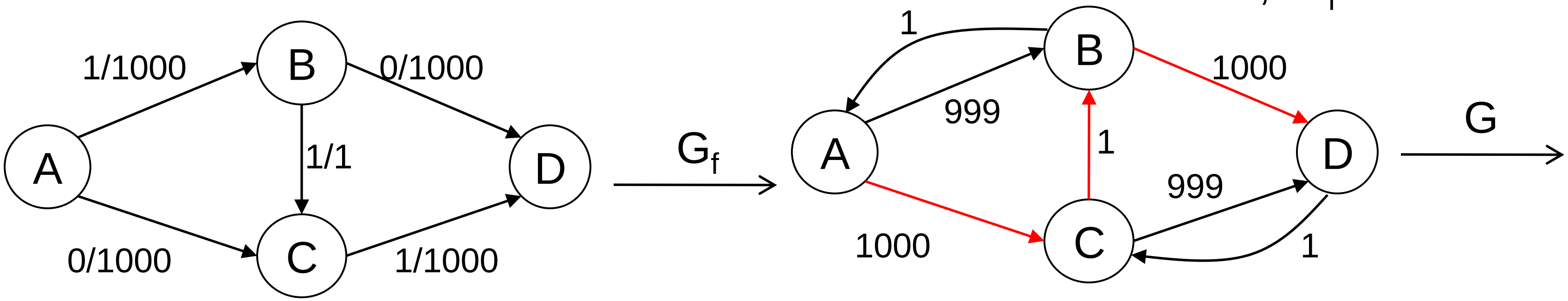
- Ford – Fulkerson(G,s,t)
  - **Pentru fiecare** (u,v) din E
    - f(u,v) = f(v,u) = 0 // O(E)
  - **Cât timp** // O(?)
    - Există o cale reziduală p între s..t în G_f // O(E)
      - $c_f(p) = \min\{c_f(u,v) \mid (u,v) \text{ din } p\}$ // O(E)
      - **Pentru fiecare** (u,v) din p // O(E)
        - $f(u,v) = f(u,v) + c_f(p)$
        - $f(v,u) = -f(u,v)$
  - **Întoarce** |f|

Complexitate?

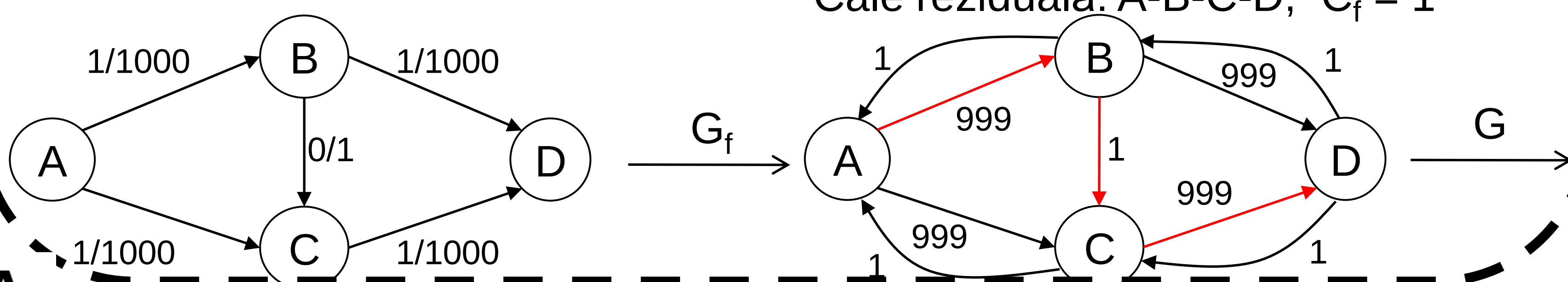
Exemplu Ford – Fulkerson (1)



Cale reziduală: A-B-C-D; $C_f = 1$

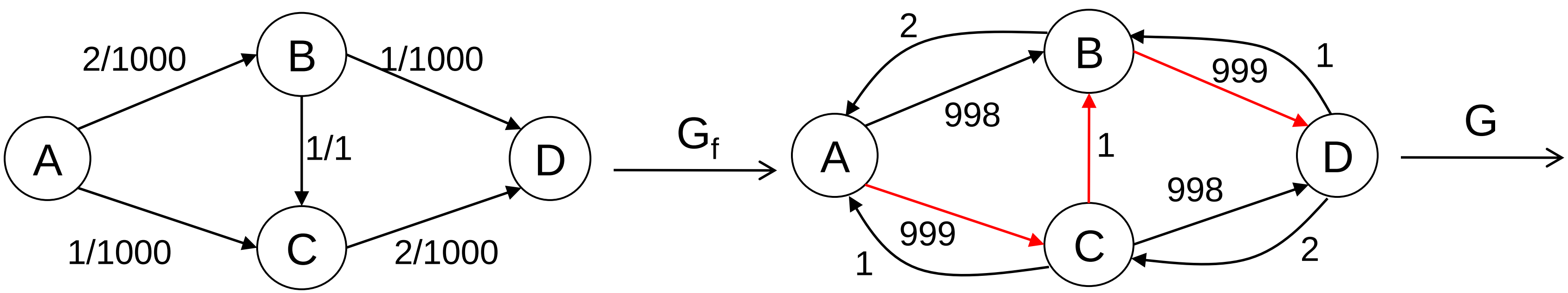


Cale reziduală: A-B-C-D; $C_f = 1$



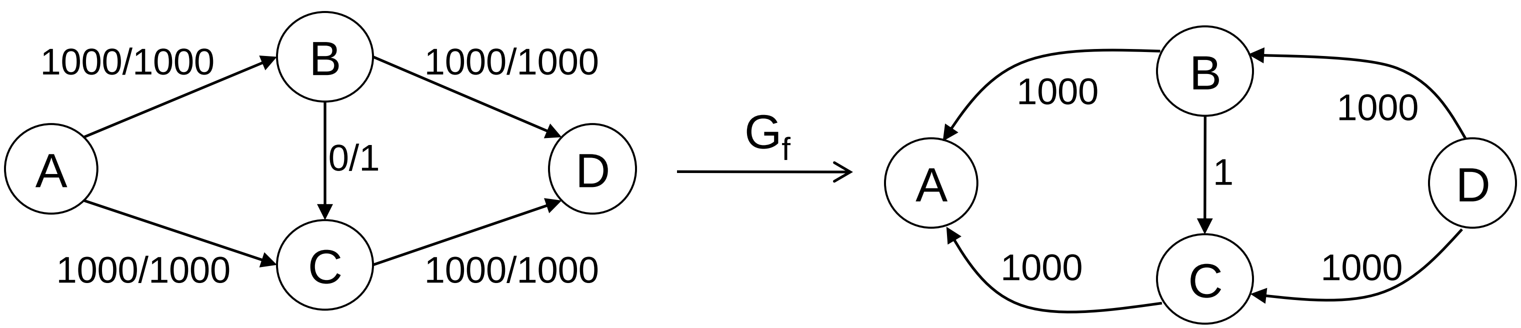
Cale reziduală: A-B-C-D; $C_f = 1$

Exemplu Ford – Fulkerson (2)



Cale reziduală: A-C-B-D; $C_f = 1$

...



Cale reziduală:Ø

După câți pași se ajunge la forma finală?

Complexitate Ford – Fulkerson

- Complexitate O(E * f_{max})
- f_{max} = fluxul maxim

Algoritmul Ford – Fulkerson – discutie

- Probleme ce pot să apară:
  - Se folosesc căi cu capacitate mică;
  - Se pun fluxuri pe mai multe arce decât este nevoie.

- Îmbunătățiri:
  - Se aleg căile reziduale cu capacitate maximă – complexitatea va depinde in continuare de f_{max} și de valoarea capacităților;
  - Se aleg căile reziduale cele mai scurte → în acest caz complexitatea nu mai depinde de f_{max} ci numai de numărul de arce (ex. Edmonds-Karp: identificarea căilor reziduale minime prin aplicarea unui BFS)