Capitolul 6
PROBLEME DE FLUXURI ÎN REȚELE

6.1. Problema fluxului maxim

Numim *rețea* (*de transport*) cu intrarea s şi ieşirea t, 4 – uplul R = (G,s,t,c) unde:

- G = (V, E) este un digraf;
- $s, t \in V, s \neq t, d_G^+(s) > 0, d_G^-(t) > 0$;
- $c : E \to \mathbf{R}_+$, c(e) este capacitatea arcului e.

Vom presupune că

$$V = \{1, 2, ..., n\} \ (n \in \mathbf{N}^*)$$

şi că

$$|E| = m.$$

Extindem funcția c la

$$c : V \times V \to \mathbf{R}_+$$

prin:

$$c((i, j)) = \begin{cases} c(ij), & ij \in E \\ 0, & ij \notin E \end{cases}$$

şi vom nota c((i,j)) = c_{ij}.

Definiție. Numim *flux* în rețeaua R = (G,s,t,c) o funcție $x : V \times V \to \mathbf{R}$ care satisface

(i) $0 \leq x_{ij} \leq c_{ij}, \quad \forall ij \in V \times V$;

(ii) $\sum\limits_{j \in V} x_{ji} - \sum\limits_{j \in V} x_{ij} = 0, \quad \forall i \in V - \{s, t\}$.

Dacă $ij \in E$ atunci x_{ij} se numeşte *fluxul* (transportat) pe arcul ij.

Evident, condiția (i) cere ca fluxul pe orice arc să fie nenegativ şi subcapacitar, iar condiția (ii) (legea de conservare a fluxului) cere ca suma fluxurilor pe arcele care intră în vârful i să fie egală cu suma fluxurilor pe arcele care ies din vârful i.

Cu convenția făcută la extensia funcției de capacitate, se observă că pentru perechile (i, j) care nu sunt arce în rețea condiția (i) impune ca fluxul să fie 0, şi evident cele doua definiții sunt echivalente.

Dacă se sumează relațiile (ii) (pentru $i \in V - \{s, t\}$) se obține:

$$0=\sum_{i\neq s,t}\left(\sum_{j\in V}x_{ji}-\sum_{j\in V}x_{ij}\right)=\sum_{i\neq s,t}\sum_{j\neq s,t}x_{ji}-\sum_{i\neq s,t}\sum_{j\neq s,t}x_{ij}+$$

$$+\sum_{i\neq s,t}x_{si}+\sum_{i\neq s,t}x_{ti}-\sum_{i\neq s,t}x_{is}-\sum_{i\neq s,t}x_{it}=-\left(\sum_i x_{is}-\sum_i x_{si}\right)-\left(\sum_i x_{it}-\sum_i x_{ti}\right),$$

Definiție. Dacă x este un flux în rețeaua R = (G,s,t,c) se numeşte *valoarea fluxului* x numărul

$$v(x)=\sum_{j\in V}x_{jt}-\sum_{j\in V}x_{tj}.$$

$v(x)$ se poate interpreta ca fiind fluxul net care ajunge în ieşirea rețelei sau (conform egalității obținute mai sus) fluxul net care iese din intrarea rețelei.

În orice rețea R = (G,s,t,c) există un flux, fluxul nul $x_{ij}=0\ \forall ij$, de valoare 0.

Problema fluxului maxim

Dată R = (G,s,t,c) o rețea, să se determine un flux de valoare maximă.

Problema se poate formula, ca o problemă de programare liniară:

$$\begin{cases}\max v\\ \\ \sum_j x_{ji}-\sum_j x_{ij}=0,\ \forall i\neq s,t\\ \sum_j x_{js}-\sum_j x_{sj}=-v\\ \sum_j x_{jt}-\sum_j x_{tj}=v\\ 0\le x_{ij}\le c_{ij}\ \ \forall ij\end{cases}$$

Definiție. Dacă P este un drum în $\overline{G}$, multigraful suport al digrafului *G*, şi $e=v_iv_j$ este o muchie a lui P atunci:

- dacă e corespunde arcului v_iv_j al lui G, e se numeşte *arc direct* al drumului P;
- dacă e corespunde arcului v_jv_i al lui G, atunci e se numeşte *arc invers*.

Definiție. Fie R = (G,s,t,c) şi x flux în R. Se numeşte *C-drum* (în R relativ la fluxul x) un drum D în $\overline{G}$ cu proprietatea că $\forall ij\in E(D)$:

$x_{ij}<c_{ij}$ dacă ij este arc direct;

$x_{ji}>0$ dacă ij este arc invers.

Dacă D este un C – drum şi $ij \in E(D)$, se numeşte *capacitate reziduală* a lui ij (relativ la C – drumul D) numărul

$$r(ij) = \begin{cases} c_{ij} - x_{ij}, & \text{ij arc direct în D} \\ x_{ji}, & \text{ij arc invers în D.} \end{cases}$$

Capacitatea reziduală a drumului D este:

$$r(D) = \min_{e \in E(D)} r(e).$$

Definiție. Se numeşte *drum de creştere* a fluxului x, în rețeaua R = (G,s,t,c), un C-drum de la s la t.

Se arată că:

Lemă.

Dacă D este un drum de creştere a fluxului x în rețeaua R = (G,s,t,c), atunci $x^1 = x \otimes r(D)$ definit prin

$$x^1_{ij} = \begin{cases} x_{ij}, & \overline{ij} \notin E(D) \\ x_{ij} + r(D), & ij \in E(D), \text{ ij arc direct în D} \\ x_{ij} - r(D), & ji \in E(D), \text{ ji arc invers în D} \end{cases}$$

este flux în R şi $v(x^1) = v(x) + r(D)$.

Rezultă că dacă x admite un drum de creştere atunci x nu este flux de valoare maximă.

Definiție. Fie R = (G,s,t,c). Se numeşte *secțiune* în rețeaua R, o partiție (S,T) a lui V cu $s \in S$ şi $t \in T$.

Capacitatea secțiunii (S,T) este

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

(suma capacităților arcelor de la S la T).

Se arată că:

Lemă.

Dacă x este un flux în R = (G,s,t,c) şi (S,T) este o secțiune a rețelei, atunci

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

(valoarea fluxului este egală cu fluxul net ce trece prin orice secțiune)

Lemă. Dacă x este un flux în R = (G,s,t,c) şi (S,T) este o secțiune, atunci $v(x) \le c(S,T)$.

Teoremă. (**Teorema drumului de creştere**)

Un flux x este de valoare maximă într-o rețea R, dacă şi numai dacă, nu există drumuri de creştere a fluxului x în rețeaua R.

Să observăm că $\forall i \in S$ şi $\forall j \in T$ avem:

- dacă $ij \in E$ atunci $x_{ij} = c_{ij}$ şi
- dacă $ji \in E$ atunci $x_{ji} = 0$

(altfel, C-drumul de la s la i se poate extinde la un C-drum de la s la j).

Deci, conform lemei precedente:

$$v(x) = \sum_{i \in S} \sum_{j \in T} (x_{ij} - x_{ji}) = \sum_{i \in S} \sum_{j \in T} (c_{ij} - 0) = c(S,T)$$

şi prin urmare x este flux de valoare maximă.

Teoremă. (Teorema fluxului întreg)

Dacă toate capacităţile sunt întregi, atunci există un flux de valoare maximă cu toate componentele întregi (flux întreg de valoare maximă).

Demonstraţie.

Fie algoritmul

```
1:   x⁰ ← 0;  i ← 0;
2:   while (∃Pᵢ drum de creştere relativ la xⁱ) do
         {
             xⁱ⁺¹ ← xⁱ ⊗ r(Pᵢ);
             i + +
         }
```

Se observă că "x^i are *componente întregi*" este un invariant al algoritmului (din definiţia lui $r(P_i)$, dacă toate capacităţile sunt întregi, rezultă că $r(P_i)$ este întreg în ipoteza că x^i este întreg) şi că la fiecare iteraţie a pasului 2 valoarea fluxului curent creşte cu măcar o unitate, deci pasul 2 se repetă de cel mult $c(\{s\}, V - \{s\}) \in \mathbf{Z}_+$ ori. Fluxul final obţinut este de valoare maximă.

Observaţie. Algoritmul, descris mai sus, este finit şi în cazul capacităţilor raţionale.

Teoremă. (Ford-Fulkerson)

Valoarea maximă a unui flux în reţeaua R =(G,s,t,c) este egală cu capacitatea minimă a unei secţiuni a reţelei.

Demonstraţie.

Dacă dispunem de un algoritm care, pornind de la un flux iniţial x^0 (x^0 există întotdeauna, de exemplu $x^0 = 0$), construieşte într-un număr finit de paşi un flux x, care nu admite drumuri de creştere, atunci secţiunea construită în demonstrația teoremei antecedente satisface împreună cu x enunțul teoremei.

Pentru cazul capacităților raționale algoritmul descris satisface această condiție.

Pentru cazul capacităților reale există un algoritm, datorat lui Edmonds şi Karp.

Algoritmul lui Ford şi Fulkerson pentru aflarea unui flux de valoare maximă

Se va folosi un procedeu de etichetare a vârfurilor rețelei, în vederea depistării drumurilor de creştere a fluxului curent x. Dacă nu există drumuri de creştere, fluxul va fi de valoare maximă.

Eticheta atribuită unui vârf $j \in V$ are trei componente (e_1, e_2, e_3) unde $e_1 \in V \cup \{0\}$, $e_2 \in \{\text{direct, invers}\}$; $e_3 \in \mathbf{R}_+$ şi au următoarea semnificație:

- dacă $e_2 =$ direct şi $e_1 = i$ atunci

$\exists$ un C-drum P de la s la j cu ultimul arc ij, arc direct şi r(P) = e_3;

- dacă $e_2 =$ invers şi $e_1 = i$ atunci

$\exists$ un C-drum P de la s la j cu ultimul arc ij, arc invers şi r(P) = e_3.

Inițial, se etichetează sursa s cu eticheta $(0, . , \infty)$. Celelalte vârfuri primesc etichetă prin "cercetarea" vârfurilor deja etichetate:

Dacă i este un vârf etichetat, atunci $\forall j \in V$:

- dacă j neetichetat,
$$ij \in E \text{ şi } x_{ij} < c_{ij}$$
atunci j se etichetează
$$e = (i, \text{direct}, \min(e_3[i], c_{ij} - x_{ij}));$$
- dacă j neetichetat,
$$ji \in E \text{ şi } x_{ji} > 0$$
atunci j se etichetează
$$e = (i, \text{invers}, \min(e_3[i], x_{ji})).$$

Evident, în acest fel se respectă semnificația celor trei componente ale etichetelor.

Numim această procedură *etichetare*(i).

Atunci când în procedura de etichetare s-a atribuit etichetă vârfului t, s-a obținut un drum de creştere P a fluxului curent, de capacitate reziduală $r(P) = e_3[t]$ şi ale cărui vârfuri se depistează în O(n) explorând prima componentă a etichetelor. Modificarea fluxului $x \otimes r(P)$ se execută în acest mers înapoi, tot în O(n).

Pentru noul flux se reia procedura de etichetare.

Dacă toate vârfurile etichetate au fost cercetate şi nu s-a reuşit etichetarea vârfului t, rezultă că fluxul curent nu admite drumuri de creştere, este deci de valoare maximă, iar dacă S = mulţimea vârfurilor etichetate atunci (S, V − S) este o secţiune de capacitate minimă.

Descrierea algoritmului

```
1.    Se alege x = (x_ij) flux inițial (de exemplu fluxul nul);
       Se etichetează s cu (0,.,∞).
2.    while (∃ vârfuri etichetate necercetate) do
        {
        "alege" un vârf etichetat şi necercetat i;
        etichetare(i);
        if (t a primit etichetă) then
              {
              modifică fluxul pe drumul dat de etichete;
              şterge toate etichetele;
              etichetează s cu (0,.,∞)
              }
        }
3.    S←{i|i are etichetă}
       T←V−S
      x este flux de valoare maximă.
      (S,T) este secțiune de capacitate minimă.
```

Complexitatea algoritmului

Pentru fiecare creştere a fluxului, sunt necesare cel mult 2m (m = |E|) inspecţii de arce în vederea etichetării.

Dacă toate capacităţile sunt întregi atunci vor fi necesare cel mult v (v = valoarea fluxului maxim) creşteri succesive. Rezultă că algoritmul are complexitatea O(mv).

Dacă U este o margine superioară a capacităţilor arcelor atunci

$$v \le (n-1)U$$

((n − 1)U este o margine superioară a capacităţii secţiunii

$$(\{s\}, V - \{s\}),$$

deci algoritmul are complexitatea

$$O(nmU).$$

Dezavantajele algoritmului sunt legate de neconvergenţa în cazul capacităţilor iraţionale (deşi practic, în implementări nu este cazul), şi de faptul ca mărimile capacităţilor influenţează comportarea sa, acestea neconstituind o măsură a volumului datelor de intrare.

6.2. Fluxuri de cost minim

Fie R = (G,s,t,c) o rețea şi x un flux de la s la t în R.

Considerăm $a : E \to \mathbf{R}$ o funcție de cost care asociază fiecărui arc $ij \in E$, $a(ij) = a_{ij}$ costul (transportului unei unități de flux) pe arcul ij.

Costul fluxului x se defineşte ca fiind

$$a(x) = \sum_{i,j} a_{ij} x_{ij} .$$

Problema fluxului de cost minim

Dată R o rețea, $v \in \mathbf{R}^+$ şi $a : E \to \mathbf{R}$ funcție de cost, să se determine x^0 flux în R astfel încât

$$a(x^0) = \min\{a(x) \mid x \text{ flux în R}, v(x) = v\} .$$

Observăm că, dacă v nu depăşeşte valoarea fluxului maxim în rețeaua R, atunci problema are întotdeauna soluții, a(x) fiind liniară, iar mulțimea fluxurilor de valoare dată v fiind mărginită şi închisă în $\mathbf{R}^m$.

Definiție. Fie x un flux în R = (G,s,t,c) şi $a : E \to \mathbf{R}$ o funcție de cost.

Dacă P este un C-drum în R relativ la fluxul x, atunci costul drumului P se defineşte

$$a(P) = \sum_{\substack{ij \in P \\ ij \text{ direct}}} a_{ij} - \sum_{\substack{ij \in P \\ ij \text{ invers}}} a_{ji} .$$

Dacă C este un C-drum închis, a(C) se calculează după aceeaşi formulă, după stabilirea unui sens de parcurgere a lui C (este posibil ca ambele sensuri de parcurgere ale lui C să satisfacă definiția unui C-drum).

1. Din definiția dată, rezultă că dacă P este drum de creştere relativ la fluxul x, atunci $x^1 = x \otimes r(P)$ este un flux de valoare $v(x^1) = v(x) + r(P)$ şi de cost $a(x) + r(P) \cdot a(P)$.

2. Dacă C este un C-drum închis relativ la x, atunci $x^1 = x \otimes r(C)$ este un flux de valoare $v(x^1) = v(x)$ şi de cost $a(x^1) = a(x) + r(C) \cdot a(C)$.

Dacă a(C) < 0 atunci x^1 este un flux de aceeaşi valoare ca şi x, dar de cost strict mai mic.

Teoremă.

Un flux de valoare v este de cost minim dacă şi numai dacă nu admite C - drumuri închise de cost negativ.

Demonstrație.

Necesitatea este evidentă din observația anterioară.

Suficiența. Fie x un flux de valoare v, care nu admite C - drumuri închise de cost negativ.

Fie x^* un flux de valoare v, de cost minim astfel încât

$$\Delta(x,x^*) = \min\{\Delta(x,x') \mid x' \text{ flux de valoare v şi cost minim}\}$$

unde $\Delta(x,x') = |\left\{ij \mid x_{ij} \neq x'_{ij}\right\}|$.

Dacă $\Delta(x,x^*) = 0$ rezultă că $x = x^*$ şi deci x este de cost minim.

Dacă $\Delta(x,x^*) > 0$, fie ij astfel încât $x_{ij} \neq x^*_{ij}$. Presupunem $0 \leq x_{ij} < x^*_{ij} \leq c_{ij}$(astfel, raționamentul este similar). Din legea de conservare a fluxurilor rezultă că

$$\exists jk \in E \text{ astfel încât } 0 \leq x_{jk} < x^*_{jk} \leq c_{jk}$$

sau

$$\exists kj \in E \text{ astfel încât } 0 \leq x^*_{kj} < x_{kj} \leq c_{jk}.$$

Repetând acest raționament, deoarece numărul vârfurilor este finit, se va obține C, un C-drum închis relativ la x în R.

Considerând sensul invers de parcurgere pe C se obține un C-drum C', închis relativ la x^*.

Deoarece a(C) $\geq$ 0 din ipoteză, iar a(C') = – a(C), rezultă, din necesitatea teoremei (x^* este de cost minim), că a(C) = 0.

Modificând fluxul x^* cu $\delta(C')$ pe C', unde

$$\delta(C') = \min\{\min_{kj \text{ direct în } C'} x_{kj} - x^*_{kj}, \min_{kj \text{ invers în } C'} x^*_{jk} - x_{jk}\}$$

se obține un flux x' cu

$$v(x') = v(x^*) = v,\ a(x') = a(x^*) + \delta(C') \cdot a(C') = a(x^*),$$

deci de cost minim, dar, cu $\Delta(x,x') < \Delta(x,x^*)$, contradicție.

Deci $\Delta(x,x^*) = 0$.

Teoremă.

Dacă x este un flux de valoare v şi de cost minim iar P_0 este un drum de creştere, astfel încât

$$a(P_0) = \min\{a(P) \mid P \text{ drum de creştere relativ la x}\}$$

atunci $x^1 = x \otimes r(P_0)$ este un flux de valoare $v(x^1) = v + r(P_0)$ şi de cost minim.

Linia demonstrației este următoarea: presupunând prin reducere la absurd că x^1 nu este de cost minim, atunci x^1 admite un C – drum închis C de cost negativ. Cum x era flux de cost minim rezultă că $E(C) \cap E(P_0) \neq \emptyset$.

Dacă $ij \in E(C) \cap E(P_0)$, atunci va rezulta că $P_0 \cup C - ij$ conține un drum de creştere relativ la x de cost mai mic decât P_0.

Un drum de creştere de cost minim poate fi depistat cu ajutorul algoritmilor de drum minim.

Dacă x este un flux în R şi $a : E \to \mathbf{R}$ este funcția de cost atunci considerând $a_{ij} = \infty$ dacă $ij \notin E$ (caz în care $x_{ij} = 0$), construim

$$\overline{a}_{ij} = \begin{cases} a_{ij}, & x_{ij} < c_{ij} \text{ şi } x_{ji} = 0 \\ \min\{a_{ij}, -a_{ji}\}, & x_{ij} < c_{ij} \text{ şi } x_{ji} > 0 \\ -a_{ji}, & x_{ij} = c_{ij} \text{ şi } x_{ji} > 0 \\ \infty, & x_{ij} = c_{ij} \text{ şi } x_{ji} = 0 \end{cases}$$

Un drum de pondere minimă de la s la t în raport cu ponderile $\overline{a}_{ij}$ corespunde unui drum minim de creştere în R relativ la fluxul x.

Un circuit de pondere negativă în raport cu ponderile $\overline{a}_{ij}$ corespunde unui C – drum închis în R relativ la x, de cost negativ.

Rezultă, următorul algoritm pentru rezolvarea problemei fluxului de cost minim.

Algoritm generic de rezolvare a problemei fluxului de cost minim

```
{
0:     Se consideră x = (x_ij) un flux cu valoarea v' ≤ v;
       {x poate fi fluxul nul sau un flux y determinat cu
        ajutorul algoritmului de flux maxim şi apoi
        considerând x = v/v(y) y }
1:     while (∃ circuite de pondere < 0 relativ la ā_ij) do
       {      determină un astfel de circuit;
               modifică fluxul pe acest circuit
       }
2:     while v(x) < v do
       {      aplică un algoritm de drum minim în raport cu
              ponderile ā_ij pentru depistarea unui C - drum
              P de cost minim;
              x ← x ⊗ min(r(P), v − v(x))
       }
}
```

Complexitatea pentru pasul 2 este O(n^3v), dacă se pleacă de la fluxul nul. Se poate dovedi că pasul 1 se poate implementa astfel ca numărul iterațiilor să fie O($nm^2 \log n$).

În continuare se prezintă o nouă descriere a algoritmului lui Ford şi Fulkerson şi se aplică acest algoritm pe un exemplu.

6.3. Algoritmul Ford-Fulkerson

Algoritmul **Ford-Fulkerson** (denumit astfel după L.R. Ford, Jr şi D.R. Fulkerson) calculează fluxul maxim într-o rețea de fluxuri. A fost publicat 1956.

Ideea algoritmului este foarte simplă: Atâta timp cât există un drum de la sursă la țintă, cu capacități disponibile pe toate muchiile drumului, vom trimite flux pe unul din aceste drumuri. Se găseşte, ulterior, un astfel de drum şi aşa mai departe. Un drum cu capacități disponibile se numeşte *drum de creştere*.

Algoritmul

Se dă un graf $G = (V,E)$, având capacitatea $c(u,v)$ şi fluxul $f(u,v)=0$ pentru muchia de la u la v. Vrem să găsim fluxul maxim de la sursa s la ținta t. După fiecare pas al algoritmului se mențin următoarele:

- $f(u,v) \le c(u,v)$. Fluxul de la u la v nu depăşeşte capacitatea.
- $f(u,v) = -f(v,u)$. Se menține fluxul net între u şi v. Dacă în realitate a unități trec de la u la v, iar b unități trec de la v la u, menținem $f(u,v) = a - b$ şi $f(v,u) = b - a$.
- $\sum_v f(u,v) = 0 \Leftrightarrow f_{in}(u) = f_{out}(u)$. Pentru toate nodurile u, exceptând s şi t. Fluxul ce intră într-un nod trebuie să fie egal cu cel ce iese din nodul respectiv.

Aceasta înseamnă că fluxul prin rețea este ceea ce numim *„flux legal”* după fiecare etapă parcursă a algoritmului. Definim **rețeaua reziduală** $G_f(V,E_f)$ ca fiind rețeaua cu capacitate

$$c_f(u,v) = c(u,v) - f(u,v)$$

şi fără flux. A se ține cont de faptul că nu se ştie cu certitudine că

$$E = E_f,$$

deoarece se poate întâmpla ca, trimițând flux pe *(u, v)* acesta să se blocheze (satureze), însă dă naştere unei noi muchii (*v*, *u*) în rețeaua reziduală.

Algoritmul Ford-Fulkerson

Date de intrare: Graful *G* cu capacitatea *c*, nodul sursă *s* şi nodul țintă *t*.

Date de ieşire: Un flux maxim *f* de la *s* la *t*

1. $f(u, v) \leftarrow 0$ pentru toate muchiile (u, v)
2. Cât timp există un drum P de la s la t, astfel încât $c_f(u, v) > 0$ pentru toate muchiile $(u, v) \in P$:
    1. Găsirea $c_f(p) = \min\{c_f(u, v) \mid (u, v) \in p\}$
    2. Pentru fiecare muchie $(u, v) \in P$
        1. $f(u, v) \leftarrow f(u, v) + c_f(p)$
           (Se trimite flux de-a lungul drumului)
        2. $f(v, u) \leftarrow f(v, u) - c_f(p)$
           (Fluxul ar putea fi „întors" ulterior)

Drumul poate fi găsit cu ajutorul, spre exemplu, metodei *breadth-first search* sau al metodei *depth-first search* în graful $G_f(V, E_f)$. Dacă se foloseşte cel dintâi, algoritmul se numeşte *Edmonds-Karp.*

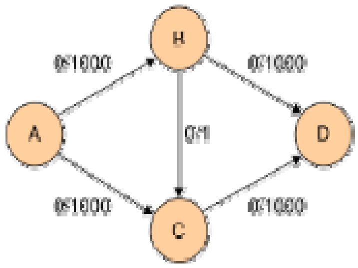
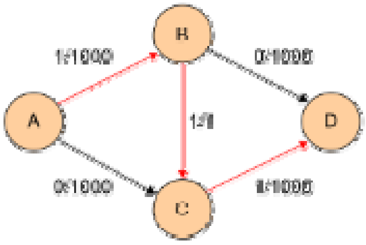
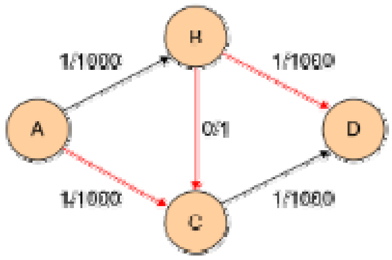
Complexitate

Fluxul maxim va fi atins în momentul în care nu vor mai fi găsite drumuri ce permit creşterea fluxului corespunzător. Cu toate acestea, nu există certitudinea unei astfel de situații, astfel că singurul lucru ce se poate garanta este acela că, odată cu parcurgerea algoritmului, rezultatul este cel corect. În cazul în care algoritmul rulează la infinit, s-ar putea întâmpla ca fluxul să nici nu conveargă către fluxul maxim. Dar această se poate întâmpla doar în cazul valorilor iraționale atribuite fluxului. Când capacitățile sunt întregi, timpul de rulare al algoritmului Ford-Fulkerson este de ordinul $O(|E| * f)$, unde |E| reprezintă numărul de muchii ale grafului, iar *f* reprezintă fluxul maxim al grafului. Acesta deoarece fiecare drum de creştere poate fi găsit într-un timp de ordinul $O(|E|)$, mărind fluxul cu o cantitate întreagă, care este cel puțin 1.

O variantă a algoritmului Ford-Fulkerson , ce garantează finalitatea şi un timp de rulare independent de valoarea fluxului maxim, este algoritmul Edmonds Karp, a cărui timp de rulare este de ordinul $O(|V||E|^2)$.

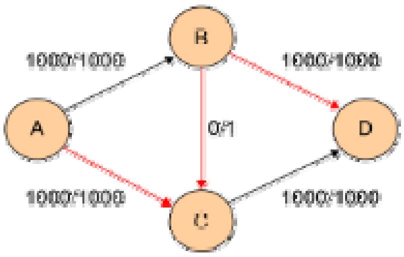
Exemplu

Următorul exemplu indică primii paşi ai algoritmului Ford-Fulkerson într-o rețea cu 4 vârfuri, *sursa* fiind A, iar *ținta* (*nodul terminal*) D. Drumurile de creştere sunt găsite cu ajutorul metodei *depth-first search* (*căutării în adâncime*), unde vecinii sunt vizitați în ordine alfabetică. Acest exemplu imaginează cel mai sumbru comportament al algoritmului. La fiecare pas, este trimis un flux având valoarea 1 de-a lungul rețelei. A se vedea că dacă s-ar fi folosit o căutare de tipul *breadth-first search*, ar fi fost nevoie de doar doi paşi.

Drum	**Capacitate**	**Fluxul în rețea rezultat**
	Fluxul inițial în rețea	
A,B,C,D	$\min(c_f(A,B),c_f(B,C),c_f(C,D))$ $= \min(c(A,B)\, f(A,B),$ $c(B,C) - f(B,C),$ $c(C,D) - f(C,D))$ $= \min(1000 - 0,$ $1 - 0,1000 - 0) = 1$	
A,C,B,D	$\min(c_f(A,C),c_f(C,B),c_f(B,D))$ $= \min(c(A,C) - f(A,C),$ $c(C,B) - f(C,B),$ $c(B,D) - f(B,D))$ $= \min(1000 - 0,$ $0 - (-1),1000 - 0) = 1$	

După încă1998 pași….

Rețeaua finală a fluxului



A se remarca modul în care fluxul este „împins înapoi” de la *C* la *B*, în momentul în care se găsește drumul *A*, *B*, *C*, *D*.