ALGORITMICA GRAFURILOR
Săptămâna 10

C. Croitoru

croitoru@info.uaic.ro

FII

December 4, 2013

OUTLINE

1. **Fluxuri** (ag 13-14 allinone.pdf pag. 239 $\longrightarrow \ldots$)

2. **Problemele pentru seminarul 10**

Problema fluxului maxim

Algoritmi de tip preflux

Se numește **preflux** în rețeaua R, o funcție $x : E \to \mathbf{R}$ astfel încît

(i) $$0 \leq x_{ij} \leq c_{ij} \quad \forall ij \in E$$

(ii) $$\forall i \neq s \ e_i = \sum_{j:ji \in E} x_{ji} - \sum_{j:ij \in E} x_{ij} \geq 0.$$

Numărul $e_i \quad i \in V - \{s, t\}$ se numește **excesul** din vîrful i. Dacă $i \in V - \{s, t\}$ și $e_i > 0$ atunci i se numește **nod activ**. Dacă $ij \in E$ x_{ij} va fi numit **fluxul pe arcul** ij.

Dacă în rețeaua R nu există noduri active, atunci **prefluxul** x **este flux** de la s la t în R de valoare e_t. *Ideea algoritmului de tip preflux*: se pornește cu **un preflux** în R și **se transformă** prin modificări ale fluxului pe arce **într-un flux** care **nu admite drumuri de creștere.**

Reprezentarea digrafului G cu ajutorul listelor de adiacență. **Totuși, vom considera că dacă $ij \in E$ atunci și $ji \in E$ (altminteri, adăugăm arcul ji cu capacitate 0).**

Problema fluxului maxim

Algoritmi de tip preflux

x preflux în R, $ij \in E$. **Capacitatea reziduală** a arcului ij este

$$r_{ij} = c_{ij} - x_{ij} + x_{ji}$$

(reprezentînd fluxul adițional ce poate fi *"trimis"* de la nodul i la nodul j utilizînd arcele ij și ji).

A *"trimite"* flux de la i la j înseamnă să creștem fluxul pe arcul ij sau să micșorăm fluxul pe arcul ji.

Se numește **C-drum** în R relativ la prefluxul x, un drum al lui G ale cărui arce au capacitatea reziduală pozitivă.

Se numește *funcție de distanță* în R relativ la prefluxul x, o funcție $d : V \to \mathbf{Z}_+$ care satisface

$(D1)$ $$d(t) = 0$$

$(D2)$ $$\forall ij \in E, r_{ij} > 0 \Rightarrow d(i) \leq d(j) + 1$$

P C-drum relativ la prefluxul x în R de la i la $t \Rightarrow d(i) \leq \lg(P)$ (arcele lui P au capacitate reziduală pozitivă și se aplică (D2)).
Rezultă că $d(i) \leq \tau_i$ (lungimea minimă a unui C-drum de la i la t).

Algoritmi de tip preflux

Fie x un preflux în R și d o funcție de distanță relativ la x. Un arc $ij \in E$ se numește **admisibil** dacă

$$r_{ij} > 0 \quad \wedge \quad d(i) = d(j) + 1.$$

Dacă R este o rețea, considerăm *inițializare* procedura care construiește în $O(m)$ un preflux x și o funcție de distanță d corespunzătoare acestuia:

procedure *inițializare*;
  for $\forall ij \in E$ **do**
    if $i = s$ **then** $x_{sj} \leftarrow c_{sj}$ **else** $x_{ij} \leftarrow 0$**;**
  $d[s] \leftarrow n$; $d[t] \leftarrow 0$;
  for $\forall i \in V - \{s, t\}$ **do** $d[i] \leftarrow 1$

Alegerea lui $d(s) = n$ are interpretarea:**"nu există C-drum de la s la t în R relativ la x"** (altfel, ar trebui ca lungimea acestuia să fie $\geq n$).

Dacă, în algoritmii de tip preflux vom păstra acest invariant, atunci cînd x va deveni flux, va rezulta că nu admite drumuri de creștere și deci x va fi de valoare maximă.

Problema fluxului maxim

Algoritmi de tip preflux

procedure *pompează* (i);
    // i este un vârf diferit de s, t
  alege $ij \in A(i)$ ij **admisibil;**
  "trimite" $\delta = \min(e_i, r_{ij})$ unități de flux de la i la j

Dacă $\delta = r_{ij}$ avem o **pompare saturată**, altfel pomparea e **nesaturată**.

procedure *reetichetare* (i);
    // i este un vârf diferit de s, t
  $d(i) \leftarrow \min\{d(j) + 1 \mid ij \in A(i) \ \wedge \ r_{ij} > 0\}$

Schema generală a unui algoritm de tip preflux este:

inițializare;
while $\exists$ noduri active în R **do**
{ selectează un nod activ i;
  if $\exists$ arce admisibile în $A(i)$
    then *pompează*(i)
    else *reetichetare*(i) }

Problema fluxului maxim

Algoritmi de tip preflux

Lemă *Algoritmul de tip preflux, de mai sus, are ca invariant "d* **este funcție de distanță relativ la prefluxul** x**".** *La fiecare apel al lui reetichetare(i), $d(i)$ crește strict.*

Lemă *Dacă pe parcursul algoritmului, i_0 este un nod activ, atunci există un C-drum de la i_0 la s, în R, relativ la prefluxul curent x.*

Corolar 1. $\forall i \in V \quad d(i) < 2n.$
Corolar 2. Numărul total de apeluri ale procedurii reetichetare este mai mic decît $2n^2$.
Corolar 3. Nr. total de pompări saturate este $\leq nm$.

Lemă (Goldberg și Tarjan 1986) *Numărul pompărilor nesaturate este cel mult $2n^2m$.*

Lemă *La terminarea algoritmului x este flux de valoare maximă.*

Problema fluxului maxim

Algoritmi de tip preflux

Algoritmul lui **Ahuja și Orlin** (1988) – cu o metodă de **scalare,** va mărgini numărul pompărilor nesaturate de la $O(n^2m)$ la $O(n^2 \log U)$.

Capacitățile sunt întregi, $max_{ij \in E}(1 + c_{ij}) = U$. Fie $\lceil log_2 U \rceil = K$.

Ideia algoritmului:

Se vor executa $K + 1$ etape. Pentru fiecare etapă p, cu p luînd succesiv valorile $K, K - 1, \ldots, 1, 0$ vor fi îndeplinite următoarele condiții:

(a) - la începutul etapei p, $\forall$ i satisface $e_i \leq 2^p$
(b) - în timpul etapei p se utilizează procedurile *pompare-etichetare* în vederea eliminării nodurilor active cu $e_i \in (2^{p-1}, 2^p]$.
Din alegerea lui K, în etapa inițială ($p = K$) condiția (a) este satisfăcută și deci, dacă (b) va fi invariant al algoritmului, după $K + 1$ etape, excesele nodurilor vor fi $\leq \frac{1}{2}$.
Dacă, toate transformările datelor vor păstra integritatea exceselor, va rezulta că excesul oricărui nod este 0, și, deci, dispunem de un flux de valoare maximă (datorită proprietăților funcției distanță $d(i)$).

Problema fluxului maxim

Algoritmul Ahuja-Orlin

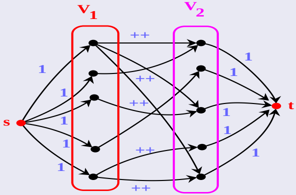
inițializare;
$K \leftarrow \lceil \log_2(U) \rceil$; $\Delta \leftarrow 2^{K+1}$;
for $p = K, K-1, \ldots, 0$ **do**
{ construiește $L(p)$; $\Delta \leftarrow \frac{\Delta}{2}$
  while $L(p) \neq \emptyset$ **do**
  { fie i primul element din $L(p)$; parcurge lista $A(i)$ din locul curent pînă se determină un arc admisibil sau se depistează sfîrșitul ei;
    if ij *este arcul admisibil găsit* **then**
    { $\delta \leftarrow \min(e_i, r_{ij}, \Delta - e_j)$;
      $e_i \leftarrow e_i - \delta$; $e_j \leftarrow e_j + \delta$;
      "trimite" δ unități de flux de la i la j;
      if $e_i \leq \frac{\Delta}{2}$ **then** șterge i din $L(p)$;
      if $e_j > \frac{\Delta}{2}$ **then** adaugă j ca prim nod în $L(p)$ }
    else // s-a depistat sfîrșitul listei
    { șterge i din $L(p)$; parcurge toată lista $A(i)$ pentru calculul lui
      $d(i) = \min\{d(j) + 1; ij \in A(i) \wedge r_{ij} > 0\}$; introdu i în $L(p)$;
      pune pointerul curent al listei $A(i)$ la început }
  }
}

Algoritmul Ahuja-Orlin

Lemă. *Numărul pompărilor nesaturate este cel mult* $8n^2$ *în fiecare etapă a scalării, deci* $O(n^2 \log U)$ *în total.*

Teoremă. (Ahuja-Orlin 1988) *Algoritmul de tip preflux cu scalarea exceselor are complexitatea* $O(nm + n^2 \log U)$.

Aflarea cuplajului maxim și a stabilei maxime într-un graf bipartit.



Dacă $x = (x_{ij})$ este un flux cu componente întregi în R atunci se observă că mulțimea de arce $\{ij \mid i \in V_1, j \in V_2 \quad \wedge x_{ij} = 1\}$ induce în graful G bipartit un cuplaj $M(x)$. În plus, $v(x)$ este cardinalul cuplajului $M(x)$.

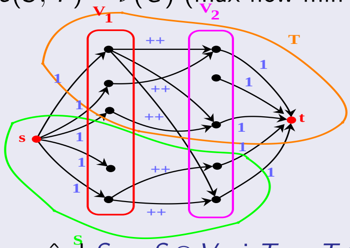
Reciproc, orice cuplaj din G induce o mulțime de arce neadiacente în G_1; dacă pe fiecare astfel de arc $ij \quad (i \in V_1, j \in V_2)$ se consideră fluxul x_{ij} egal cu 1 și de asemenea $x_{si} = x_{jt} = 1$, și luînd fluxul $x = 0$ pe orice alt arc, atunci fluxul construit are valoarea $|M|$.

Rezolvînd problema fluxului maxim pe rețeaua R se determină (pornind de la fluxul nul) în $O(nm + n^2 \log n)$ un cuplaj de cardinal maxim în graful bipartit G.

Aplicații combinatorii

Aflarea cuplajului maxim și a stabilei maxime într-un graf bipartit.

Fie (S, T) secțiunea de capacitate minimă ce se obține în $O(m)$, din fluxul maxim aflat. Avem, $c(S, T) = \nu(G)$ (max-flow min-cut).



Cum $\nu(G) < \infty$, rezultă că punînd $S_i = S \cap V_i$ și $T_i = T \cap V_i$ $(i = 1, 2)$, avem: $|T_1| + |S_2| = \nu(G)$, iar $X = S_1 \cup T_2$ este **mulțime stabilă** în graful G (pentru a avea $c(S, T) < \infty$).
În plus, $|X| = |V_1 - T_1| + |V_2 - S_2| = n - \nu(G)$.

Rezultă că X este stabilă de cardinal maxim, întrucît $n - \nu(G) = \alpha(G)$ (teorema lui König).

Problemele pentru seminarul 10

Se vor discuta (cel puțin) patru probleme dintre următoarele:

1. **Problemele 4,2 Setul 11**
2. **Problemele 1,4 Setul 16**
3. **Problema 2, Setul 15**
4. **Problema 2 Setul 18**
5. **Problema 1, Setul 19**
6. **Problema 2 Setul 22**