Simionesei Loredana-Marinela
Prisacaru Elena-Cătălina
grupa A5 anul 2

Tema 3
Algoritmica Grafurilor

Problema 2

a) Reţeaua construită este R(D,c,s,t) cu următoarele proprietăţi:

1. $V(R)=V(G) \cup \{s,t\}$
2. $E(R)=\{(u,v),(v,u)$ arce, pentru orice muchie $\{u,v\}\in E(G)\} \cup \{(s,i)$, pentru orice $i \in V(G)\} \cup \{(i,t)$, pentru orice $i \in V(G)\}$
3. $c(i,j)=1$ pentru orice $(i,j) \in E(R)$, $c(i,j)=m$, $c(i,t)=m+2g-d_G(i)$, $\forall\, i \in V(R)$ şi $c(i,j)=0$ dacă mai există şi alte arce care nu aparţin lui E(R).

Împărtind mulţimea de vârfuri din reţea V_R în două mulţimi S şi T ($s \in S$ şi $t \in T$), se observă o tăietură s-t.
Considerăm $V_1=S-\{s\}$, $V_2=T-\{t\}$.
$V_1=\emptyset \Rightarrow c(S,T)=m|V(R)|$. Astfel,
$c(S,T)=\sum_{i\in S,\, j\in T} c_{ij}=\sum_{j\in V2} c_{sj}+\sum_{i\in V1} c_{it}+\sum_{j\in V2,\, j\in V1} c_{ij} =$
$= m|V_2|+(m|V_1|+2g|V_1|-\sum_{i\in V1} d_{G(i)})+\sum_{i\in v1, j\in v2} c_{ij}=$
$=m|V(R)|+2|V_1|(g - \frac{\sum_{i\in V1} dG(i)-\sum_{i\in V1, j\in V2} 1}{2|V1|})$ (1)
Dar noi ştim că $\frac{\sum_{i\in V1} dG(i)-\sum_{i\in V1, j\in V2} 1}{2|V1|}$ reprezintă numărul de muchii ale subgrafului G' construit numai din vârfuri din V_1 în G. (2)
Din (1) şi (2) $\Rightarrow f(v_1)=\frac{\sum_{i\in V1} dG(i)-\sum_{i\in V1, j\in V2} 1}{2|V1|}$, adică
$c(S,T)=m|V(R)|+2|V_1|(g-f(V_1))$.
Conform teoremei de flux maxim de tăietură minimă, avem că valoarea fluxului maxim este egală cu valoarea tăieturii minime, adică de capacitate minimă.
Considerăm această tăietură ca fiind $C_{min.}$Avem din ipoteză faptul că $C_{min}< mn=m|V(R)|$.

Dacă $V_1=\emptyset \Rightarrow C=m|V(R)|$, adică nu se întâmplă ceea ce dorim noi.
Dacă $V_1\neq\emptyset \Rightarrow m|V(R)|>m|V(R)|+2|V_1|(g-f(V_1))$ / $-m|V(R)| \Rightarrow$
$\Rightarrow 0>2|V_1|(g-f(V_1)) \Rightarrow f(V_1)>g$ şi G' subgraf indus al lui G, unde V_1 mulţimea vârfurilor din G'.

b) Fie orice graf $G=(V,E)$, $|V|=n$ şi $|E|=m$.

Codomeniul funcţiei f pentru un astfel de graf este: $\frac{a}{b}$, $0\leq a\leq m$ şi 1, $1\leq b\leq n$. Adică pentru orice graf $0\leq f(G)\leq m$.
<u>Problema P</u>: problema determinării unui subgraf G* ai lui G astfel încât
$f(G^*)=\max f(G')$, unde G' subgraf al lui G.
Presupunem prin reducere la absurd că G^0 nu este o soluţie a problemei P.
Considerăm atunci un subgraf G^1 al lui G, G^1 este soluţie a problemei P.

Observaţie! Problema P sigur are soluţie, deoarece există un subgraf al grafului G astfel încât fracţia $\frac{numărul de muchii ale subgrafului (m)}{numărul de vârfuri ale sugrafului (n)}$, să fie maximă.
Avem:

$f(G^0)=\frac{m0}{n0}$
$f(G^1)=\frac{m1}{n1}$
————————————(-)
$dif=\frac{m0}{n0} - \frac{m1}{n1}= \frac{m0n1-m1n0}{n0n1}$.

În cazul în care am avea $n^0=n^1$ => | dif | $=|\frac{m0}{n0} - \frac{m1}{n1}|= |\frac{m0-m1}{n0}|\geq \frac{1}{n}$ deoarece diferenţa între n^0 şi n^1 este măcar 1, astfel ar fi acelaşi subgraf (adică trebuie să difere prin măcar o muchie) => contradicţie (3)
În cazul în care avem, $n^0 \neq n^1$=> $n^0n^1 \leq n(n-1)$=>$|dif| \geq \frac{1}{n(n-1)}$ =>contradicţie. (4)
Din (3) şi (4) => G^0 este soluie pentru problema P.

c) function Rezolvă_Problema_P()

```
{
st=0;
dr=0;
V(G')= Ø;
while(dr-st≥ 1/(n(n-1)))
  {
g=(st+dr)/2;
}
Construieşte reţeaua R=(V(R),E(R)); Găseşte fluxul maxim (S,T) un reţeaua R
if S={s}
  {
  dr=q;
}
   else{
     V(G')=S-{s};
}
   return subgraful G';
}
```

Algoritmul de mai sus are complexitatea $O(A(n,n+m) \log_m)$ deoarece instrucţiunea while se execută în timp logaritmic. De fiecare dată când se intră în while se construieşte o reţea cu n+2 vârfuri şi 2m+2n muchii (ca în ipoteză cu O(n) vârfuri şi O(m+n) muchii). Pentru găsirea fluxului maxim într-o reţea este complexitatea O(m,n+m).

Problema 1

1. Pentru că fiecare v ∈ V apare de $d_G(V)$ în V şi |V| se obţine din $\frac{dG(V)}{dGV)}$=1, este evidentă inegalitatea |S|≤ |T|. (1)

Dar în ipoteză avem că |S|≥ |T|. (2)
Din (1) şi (2) rezultă că |S|=|T|.
Dacă un graf G este bipartit înseamnă că orice componentă conexă a grafului G este bipartită. Pentru a demonstra că fiecare componentă conexă a lui G este un graf (bipartit) regulat, vom arăta că pentru ∀ 2 noduri s ∈ S şi t ∈ T astfel încât st ∈ E, |S|=|T| atunci $d_G(s)= d_G(t)$.

Ne vom folosi de lungimea bipatiţiilor pentru a demonstra inclusiv faptul că d_G(s)= d_G(t).
Din |S|=|T|=1 şi G graf bipartit fără vârfuri izolate (din ipoteză) => că d_G(s)= d_G(t)=1, unde s $\in$ S şi t $\in$ T.
Presupunem că pentru |S|=|T|=k, pentru fiecare muchie st $\in$ E (unde s $\in$ S şi t $\in$ T) are loc egalitatea d_G(s)= d_G(t) şi demonstrăm că egalitatea este adevărată şi pentru |S|=|T|=k+1.
Presupunem prin reducere la absurd că $\exists$ 2 vârfuri x $\in$ S şi y $\in$ T, xy $\in$ E astfel încât d_G(s) $\neq$ d_G(t).
Însă noi ştim că:
|E|=$\sum$ s$\in$ S d_G(s)=$\sum$ s$\in$ S, s$\neq$x d_G(s)+ d_G(x) (3).
|E|=$\sum$ t$\in$ T d_G(t)=$\sum$ t$\in$ T, t$\neq$y d_G(t)+ d_G(y)(4).
Graful G fără vârfurile x şi y este bipartit, S-x şi T-y sunt bipartiţii de lungime k => pentru fiecare muchie st, s $\in$ S -x şi t $\in$ T-y are loc d_G(s)= d_G(t).(5)
Din (3), (4), (5) => d_G(s)= d_G(t) => contradicie.
În concluzie, pentru orice două vârfuri v_1 şi v_2 din G între care există muchie, avem d_G(v1)= d_G(v2)=>fiecare componentă conexă a lui G este graf (bipartit) regulat.

b)

Mai sus am demonstrat că fiecare componentă conexă a grafului G este un grad K-regulat. Conform teoremei, fiecare graf k-regulat bipartit, conex are un cuplaj perfect.
Demonstraţie:
Considerăm un graf bipartit conex k-regulat cu bipartiţia (S,T). Fie $Vf_1 \subseteq$ S şi m numărul de muchii cu un capăt în Vf_1.
Pentru că fiecare v $\in$ Vf are gradul K => k|Vf|=m.
Analog, orice vârf din Vf_2(mulimea de vecini ai nodurilor din Vf_1) are gradul k => m$\leq$ k|Vf_2| =>|Vf_1|$\leq$ |Vf_2|. Din teorema lui Hall => că există un cuplaj ce saturează toate vârfurile lui S. Analog, există un cuplaj ce saturează vârfurile lui T => există un cuplaj perfect în G, pentru că |S|=|n|.
Din G este k-regulat şi |E|=$\sum_{s\in}$ S $d_{G(s)}$ =$\sum_{t\in}$ T $d_{G(t)}$ => k|S|=k|T| => |S|=|T| deoarece orice nod din S şi orice nod din T au gradul k.
Vom demonstra relaţiile |E|=$\sum_{s\in}$ S d_G(s) =$\sum_{t\in}$ T d_G(t) prin inducţie.
Baza:
|E|=1 => există o muchie intre S şi T, unde fiecare suma a gradelor este egal cu 1, adică sunt egale.
Pas inductiv:
Presupunem că pentru |E|=k avem egalitatea celor doue sume:
$\sum$ s$\in$ S d_G(s) =$\sum$ t$\in$ T d_G(t)=k.
Demonstrăm că pentru |E|=k+1 avem graf bipartit. Dacă ştergem o muchie, avem |E|=k, care este graf bipartit. Mai ştim că muchia ştearsă este în ambele situaţii egala cu 1, iar prin adăugarea ei, de unde obţinem egalitatea.
Din a) şi din ce s-a demonstrat la b) => faptul că graful G este un cuplaj perfect.

Problema 3

1. Orice digraf admite o mulţime 2-hops stabilă.

Demonstraţie:
Construim D_L=(V,E_L) şi D_H=(V,E_H) şi mulţimile de arce definite astfel:

E_L={ij ∈ E|i<j} şi E_H={ij ∈ E|j<i}. Digrafurile D_L i D_H sunt aciclice şi în fiecare există o mulţime stabilă de noduri între care nu există arce şi din fiecare nod din această mulţime există un arc direct în celelalte noduri din afara ei. Atunci fie M_L această mulţime stabilă pentru $D_{L.}$Construim M_H o mulţime stabilă pentru subdigraful lui D_H indus de $M_{L.}$Deoarece a_L este stabilă în D_L atunci pentru orice nod v ∈ V − M_L există un nod în u ∈ V − M_H astfel încât să existe arcul direct uv. De asemenea, pentru orice nod din ∈ V − M_H se ajunge din M_H prin drum de lungime cel mult 1. M_H este 2-hops stabliă în D.
Problema noastră se reduce la contrucţia lui D_L şi a lui D_H şi găsirea unei mulţimi stabile în fiecare din cele două dgrafuri aciclice.
O mulţime stabilă într-un digraf aciclic se poate găsi în timp liniar O(m+n) astfel:

```
DFS(x)
{
viz[x]=1;
for (y ∈ N⁺(x))
    if(viz[y]==0)
        DFS(y);
    TopoSort.push(x);
}
MulţimeStaăilă(D)
{
S= ∅;
for( y ∈ V(D))
    viz[y]=0;
TopoSort=∅;
for(y ∈ V(D))
    if(viz[y]=0)
        DFS(y);
for(y ∈ V(D))
    viz2[y]=0;
for(y ∈ TopoSort)
{
    if(viz2[y]=0)
        S.push(y);
    viz2[y]=1;
    for(x ∈ N⁺(y))
        viz2[x]=1;
}
return r;
}
```

<u>Corectitudine:</u>
Funcţia MulţimeStabilă returnează pentru digraful aciclic D=(V,E) o mulţime stabilă deoarece sortându-se mai întâi nodurile topologic, pentru fiecare arc xy_∈ E, x apare înaintea lui y şi în S se pune nodul x fără vecinii săi. Procedeul se repetă pentru fiecare nod încă nevizitat cu viz2. Mulimei 2-hops stabilă se construieşte conform celor spuse la început.
<u>Complexitate timp:</u>

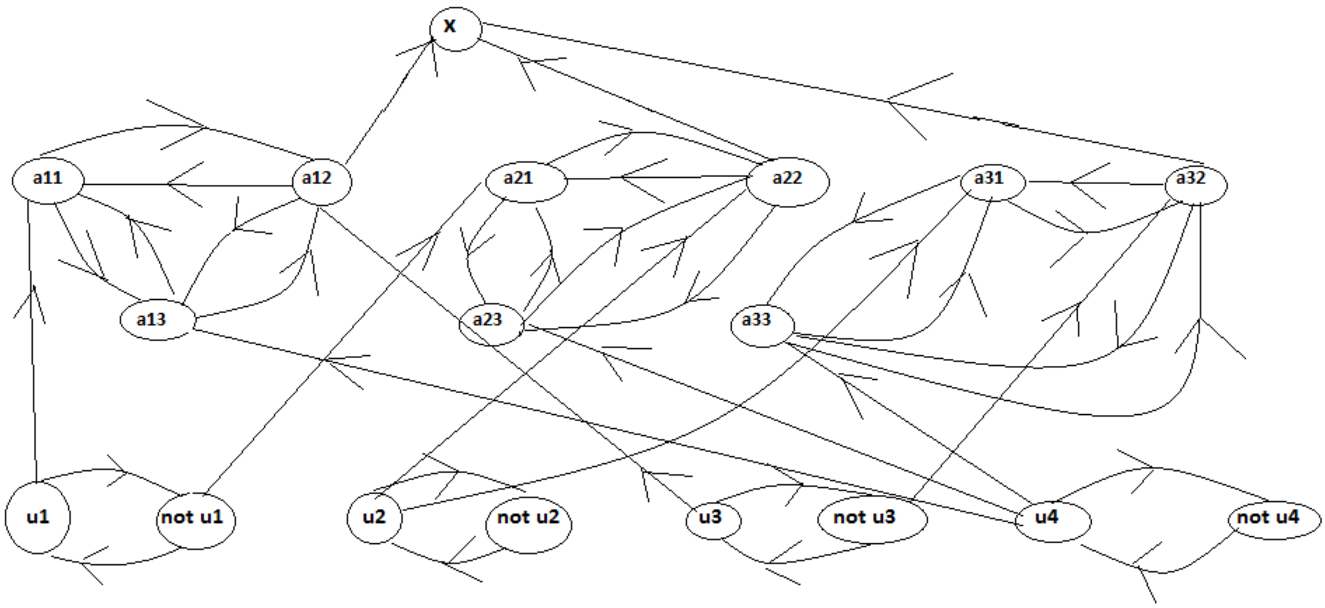
Algoritmul de găsire a unei mulţimi 2-hops stabile prezentat are complexitatea O(m+n). Construcţia celor două digrafuri e_L şi D_u se face în timp O(m+n) dacă digraful D este prezentat prin lista de adiacenţă. Construcţia unei mulţimi stabile în fiecare astfel de digraf se face ît timp lîniar, deoarece complexitatea sortării topologice este O(m+n) şi apoi o parcurgere a nodurilor se realezează în O(n) şi pentru fiecare nod incep nevizitat se mai vizitează şi vecinii, deci în total O(m+n).
Algoritmul prezentat are complexitatea O(m+n).

b)

2-hops stabilă este o problemă de decizie:
Instanţă: D=(V,E) digraf v $\in$ V
Întrebare: Există în D o mulţime 2-hops stabila care să conţină pe v?
Vom demonstra în continuare faptul că 3-SAT se reduce polinomial la problema testării dacă într-un digraf dat există o mulţime 2-hops stabilă care să conţină un vârf dat al digrafului, adică 3-SAT α 2-hops stabilă.
Considerăm U={$u_{1,...,un}$}, (unde n $\in$ N*), C=C_1ˆ C_2ˆ... ˆ C_m (unde m $\in$ N*) şi C_i=$V_{i1}$$V_{i2}$ $V_{i3,}$ unde $\forall$ i=1,m (unde pentu $\forall$ V_{ij} $\exists$ $\alpha \in$ {1,...,n} astfel încât v_{ij}=u$\alpha sau v_{ij}$=not).
Acestea reprezintă datele unei instanţe pentru problema 3-SAT.
Vom încerca acum să construim în timp polinomial în raport cu n+m un digraf D ţinând cont de faptul că o atribuire pentru variabilele booleene din U astfel încât formula C să fie adevărată <=> faptul că în digraful nostru D există o mulţime 2-hops stabilă care să conţină vârful dat al digrafului.
Pentru a construi digraful D vom urma paşii:

1. Pentru $\forall$ i $\in$ {1,...,n} vom considera diagrafuri disjuncte formate din perechi de arce simetrice A_i=({a_i, not a_i},{a_i not $a_{i,}$ not $a_i a_i$}).
2. Pentru $\forall$ j $\in$ {1,...,n} vom considera digrafuri disjuncte formate din perechi de arce simetrice B_j=({b_{j1},$b_{j2,}$$b_{j3}$}, {$b_{j1,}$$b_{j2}$},{$b_{j2,}$$b_{j1}$},{$b_{j1,}$$b_{j3}$},{$b_{j3,}$$b_{j,}$}), {$b_{j2,}$$b_{j3}$},{$b_{j3}$$b_{j2}$}).
3. Pentru $\forall$ j $\in$ {1,...,n} considerăm mulţimea de arce E={$v_{j1}$$b_{j1,}$$v_{j2}$$b_{j2}$,$j_{v3}$$b_{j3}$}, unde v_{j1} v_{j2} v_{j3} este factorul $C_{j.}$
4. Dacă X este vârful dat din digraf, vom forma arce între nodul x cu fiecare triunghi de arce creat spre x.

Luăm exemplul din curs:
U={u_1,u_2,u_3,u_4
C={u_1 or u_3 or u_4}ˆ{not u_1 or u_2 or u_4}ˆ{u_2 or not u_3 or u_4}
x - nodul dat din digraf.



"=>" Presupunem că răspunsul pentru problema 3-SAT este da, adică există o atribuire pentru variabilele booleene din U astfel încât formula C este adevărată.
Vom demonstra acum că pentru digraful D de mai sus şi un vârf x fixat uăspunsul la problema 2-hops stabilă este da (adică exista o mulţime 2-hops stabilă în D care să-l conţină pe x).
Considerăm M={u_i|u_i=adevăuat}∪{not m_i|u_i=fals} mulţimea literalilor adevăraţi. Pentru un i fixat în mulţimea M nu avem cum să regăsim atât u_i cât şi not u_i. Deci, formula C este adevărata, pentru că în fiecare clauză exista măcar un literal adevărat, ducându-se în triunghiul B_j prin drumuri de lungime≤ 2.
Fiecare nod din digrafurile A_i vor fi ori în M ori vor fi adiacente cu un nod din M (drum de lungime 1).
Se vede că x nu este adiacent cu nici un vârf din digrafurile A_i. Între vârfurile din digrafurile A_i şi X pot exista drumuri dar doar de lungime 3. Deci îl vom pune pe x în mulţimea M, acesta rămânând în continuare stabilă.
Aadar, M-mulţime 2-hops stabilă ce îl conţine şi pe x, adică răspunsul la problema 2-hops stabilă este da.
"<=" Acum vom presupune că răspunsul la 2-hops stabilă pentru instanţa D şi x fixat este da.
Considerăm M o mulţime 2-hops stabilă ce conţine pe x. De aici obţinem faptul că M poate avea cel mult un vârf din orice V(A_i) şi cel mult un vârf din orice V(B_j) (i=1,m; j=1,m).
Deci un literal va fi adevărat dacă M∩ (A_i)={u_i} şi fals dacă M ∩ (A_i)={not u_i}.
Observăm că x este inclus în mulţimea M. Dar nu există nici un drum de lungime $\leq$ 2 la vreun nod din A_i sau B_j. Pentru că în fiecare ajungem doar prin not u_i, înseamnă că u_i este în M sau între u_i şi măcar un nod din M există drum de lungime $\leq$ 2. Adică orice clauză conţine măcar un literal adevărat din M, de unde obţinem imediat faptul că formula C este adevărată. Aşadar, răspunsul la problema 3-SAT este da.