

Universitatea “Alexandru Ioan Cuza” din Iasi
Facultatea de Informatica

Algoritmica Grafurilor - TEMA 2

Studenti: Buterchi Andreea, Racovita Madalina-Alina
Grupa: B2 **Anul:** II

1 Problema 1

a) Se numeste arbore partial al lui G , un graf partial $T = (V, E')(E' \subseteq E)$ care este aciclic si conex. Arborele partial, care are suma costurilor muchiilor minima si care cuprinde toate nodurile si doar o parte din muchii, poarta numele de arbore partial de cost minim. Consideram T^* ca fiind arborele partial de cost minim al lui G , dat in raport cu functia $c : E \rightarrow \mathbb{R}$ si $\epsilon > 0$, un numar real pozitiv oarecare.

Daca T^* este arbore partial de cost minim atunci $\sum_{e \in E(T^*)} c(e) \leq \sum_{e \in E(T)} c(e), \forall T$ - arbore partial al grafului G . Vom calcula costul arborilor partiali ai grafului G in raport cu functia de cost $\bar{c} : E \rightarrow \mathbb{R}$. Prin urmare, $\bar{c}(T) = \sum_{e \in E(T)} \bar{c}(e) = \sum_{e \in E(T)} c(e) - \sum_{e \in E(T) \cap E(T^*)} \epsilon$. Alegem in mod arbitrar un arbore A apartenent al multimii de arbori partiali ai grafului G si vom demonstra ca T^* este unicul arbore partial de cost minim in raport cu functia \bar{c} . Vom calcula costurile celor doi arbori in raport cu functia $\bar{c} : E \rightarrow \mathbb{R}$. T^* are $n - 1$ muchii, lucru rezultat prin prisma definitiei arborelui, el cuprinzand toate nodurile grafului G . Astfel costul total al lui T^* va fi $\bar{c}(T^*) = \sum_{e \in E(T^*)} \bar{c}(e) = \sum_{e \in E(T^*)} (c(e) - \epsilon) = (\sum_{e \in E(T^*)} c(e)) - (n - 1) \cdot \epsilon$.

Pe de alta parte, $|E(A) \cap E(T^*)| = n - 1$, daca $A = T^*$ si $|E(A) \cap E(T^*)| < n - 1$, daca $A \neq T^*$ deoarece arborele A , difera de arborele T^* prin cel putin o muchie. Deci daca avem muchia $e \in E(A)$ cum $e \notin E(T^*) \rightarrow \bar{c}(e) = c(e)$. Ne va rezulta de aici ca in arborele A , desi are tot $(n - 1)$ muchii, exista minim o muchie al carei cost va ramane neschimbat prin functia \bar{c} deci costul maxim al lui A ar putea fi $(\sum_{e \in E(A)} c(e)) - (n - 2) \cdot \epsilon$. Si cum $\sum_{e \in E(T^*)} c(e) \leq \sum_{e \in E(A)} c(e), \forall A$ -arbore partial al grafului G (deoarece, din ipoteza, T^* este arbore partial de cost minim), ne va rezulta astfel ca $\bar{c}(T^*) < \bar{c}(A), \forall A$.

Prin urmare, T^* este unicul arbore partial de cost minim al lui G , in raport cu functia $\bar{c} : E \rightarrow \mathbb{R}$, acest avand costul mai mic decat costul oricarui alt arbore partial al grafului G .

b) Algoritmul lui Kruskal este un algoritm fundamental pentru determinarea arborilor partiali de cost minim, acesta bazandu-se pe o strategie Greedy. Pentru a determina un arbore partial de cost minim se considera initial ca nici o muchie nu este selectata. Cu alte cuvinte, se pleaca de la o padure formata din n arbori (unde $n = |V(G)|$) fiecare arbore fiind format dintr-un singur varf. La fiecare pas al algoritmului se selecteaza o muchie de cost minim care nu a mai fost selectata si care nu formeaza cicluri cu cele deja selectate.

Pentru a extrage minimul se sorteaza muchiile crescator dupa cost si se parcurg secvential muchiile ordonate iar pentru a testa daca o muchie formeaza cicluri cu muchiile deja selectate este suficient sa se testeze daca extremitatile muchiei se gasesc in aceeasi componenta conexa (adica in acelasi arbore). Algoritmul cerceteaza in cel mai defavorabil caz toate cele m muchii pentru a selecta cele $n - 1$ care nu formeaza cicluri.

Consideram ordonarea muchiilor pentru algoritmul lui Kruskal in raport cu functia $\bar{c} : E \rightarrow \mathbb{R}$, T^* fiind un arbore partial de cost minim oarecare al lui G returnat de algoritm. Conform subpunctului **a)**, in raport cu functia de cost \bar{c} , T^* este unicul arbore partial de cost minim, prin urmare, algoritmul lui Kruskal va returna arborele T^* . Observam ca in cazul unei alegeri cat mai triviale ale lui ϵ ($\epsilon \rightarrow 0$), ordonarea precedenta ramane o ordonare valida si in raport cu functia c (deoarece in acest caz $\bar{c}(e) \cong c(e), \forall e \in E(T^*)$). Aceasta este ordonarea pentru care algoritmul va returna T^* , unde T^* este un arbore partial de cost minim oarecare. **(1)**

Daca in schimb, pornim de la arborele T^* returnat de algoritmul lui Kruskal ca fiind un arbore partial de cost minim, in raport cu functia $c : E \rightarrow \mathbb{R}$, vom demonstra ca trecerea la functia de cost $\bar{c} : E \rightarrow \mathbb{R}$, va determina de asemenea existenta unei ordonari a muchiilor pentru care algoritmul sa returneze T^* . Fie $T^* = (t_1, t_2, \dots, t_{n-1})$ muchiile arborelui rezultat in urma aplicarii algoritmului, in ordinea selectarii lor. Cand facem trecerea de la functia de cost c la functia \bar{c} , algoritmul lui Kruskal va returna arborele $A = (a_1, a_2, \dots, a_{n-1})$ unde $\bar{c}(a_i) = c(t_i) - \epsilon$. Alegem $\epsilon = \min(c(e_j) - c(e_i))$ unde $j > i, i, j = \overline{1, m}$ si $c(e_j) - c(e_i) \neq 0$, considerand $E(G) = (e_1, e_2, \dots, e_i, \dots, e_j, \dots, e_m)$ fiind sortarea tuturor muchiilor grafului G in functie de cost (ϵ = cea mai mica diferenta de cost dintre doua muchii cu costuri diferite). Faptul ca introducem functia \bar{c} va modifica costurile muchiilor $e, \forall e \in E(T^*)$, restul costurilor raman neschimbate, iar acest lucru va modifica sortarea realizata

in incipitul algoritmului Kruskal pentru ordonarea crescatoare a costurilor muchiilor. Deci daca avem in sortare minim 2 muchii cu cost egal, e_i si e_{i+1} , iar $e_{i+1} \in E(T^*)$, prin introducerea functiei \bar{c} deci implicit prin scaderea lui ϵ din costul ce ii fusese asignat muchiei e_{i+1} ce face parte din arborele partial de cost minim T^* , e_{i+1} va aparea in noua sortare initiala din Kruskal inaintea lui e_i . Cu toate acestea, muchia e_{i+1} va si selectata si va face parte din ordonarea finala a muchiilor arborelui partial de cost minim returnat de Kruskal.

In ceea ce priveste secventele de muchii din sortarea initiala care au costuri diferite, muchiile apartenente arborelui T^* , vor avea oricum costul mai mic, $\bar{c}(e) = c(e) - \epsilon < c(e), \forall e \in E(T^*)$, deci algoritmul lui Kruskal va selecta mereu muchia a_i , unde $a_i \in E(T^*)$ in loc de a_j unde $a_j \notin E(T^*)$ si unde a_j este muchia imediat urmatoare cu cost diferit de a_i din sortarea initiala, $\forall i, j \in \overline{1, m}$. Si cum algoritmul lui Kruskal selecteaza muchii de cost minim, ne va rezulta tot arborele T^* . (2)

Din (1),(2), pentru orice arbore partial de cost minim T^* al lui G in raport cu functia de cost c , exista o ordonare a muchiilor grafului G pentru care algoritmul lui Kruskal returneaza T^* .

2 Problema 2

Fie $D = (V, E)$ un digraf tare conex de ordin n , iar $a: E \rightarrow \mathbf{R}$ o functie de cost pe arcele sale. Vom considera $A_k(v)$ costul minim al unui drum de lungime k de la s la v in digraful \mathbf{D} , unde s este un nod fixat din \mathbf{V} si $k \in \mathbf{N}$. Notam cu $a(X)$ costul lui X , adica suma costurilor arcelor lui X .

a) Considerand ca in digraful \mathbf{D} nu exista circuite de cost a negativ, dar exista un circuit \mathbf{C} cu $a(C) = 0$, vom demonstra ca exista un varf $v \in \mathbf{V}(\mathbf{D})$, astfel incat: $A_n(v) = \min\{a(P) | P \text{ drum de la } s \text{ la } v \text{ in } D\}^*$.

Pentru a identifica drumurile minime de la nodul de start s la v , $\forall v \in V(G)$ ne vom folosi de algoritmul lui **Bellman-Ford-Moore**. Algoritmul amintit se pliaza pe cerinta problemei, intrucat acesta genereaza drumurile minime de la un nod sursa la toate nodurile dintr-un graf dat, unde costurile asociate muchiilor pot avea valori reale, respectiv $a(C) \geq 0, \forall C \in G$, unde C reprezinta un circuit in graful G .

Problema se rezuma la demonstrarea faptului ca macar pentru unul dintre nodurile din graful G exista un drum P de lungime n de la nodul sursa s la nodul vizat, astfel incat costul drumului identificat sa fie egal cu valoarea costului drumului minim catre respectivul nod. Asadar, ca punct de plecare al demonstratiei, presupunem ca nu exista niciun nod pentru care are loc proprietatea de mai sus.

Relatiile de care ne vom folosi in cadrul demonstratiei fac parte din cadrul algoritmului **Bellman-Ford-Moore**:

$$u_{0,s} = 0; u_{0,i} = \infty; u_{k+1,i} = \min(u_{k,i}, \min_{j \in \text{incident}(i)} (u_{k,j} + a_{ji})), \forall i \neq s$$

Pentru determinarea drumului de cost minim ne vom raporta la urmatoarea relatie:

$$w_{0,s} = 0; w_{0,x} = \infty; w_{k+1,x} = \min_{y \in \text{incident}(x)} (w_{k,y} + a_{yx}), \forall x \neq s$$

Intr-un graf orientat se defineste un **drum** ca fiind o succesiune de noduri care au proprietatea ca, oricare ar fi doua noduri succesive, ele sunt legate printr-un arc. Un *drum de lungime minima* este un drum care contine numarul minim de arce din multimea nevida de drumuri dintre 2 noduri prestabilite. O definitie alternativa este aceea conform careia un drum este un mers in care muchiile sunt distincte. Luand in considerare faptul ca ne aflam in digraful D , repetarea unor arce in parcurgere, ne indica in mod concret ca ne aflam intr-un **circuit**. Avand in vedere ca valoarea costului unui circuit este $0(a(C) = 0)$, este evident ca un mers cu circuit va avea costul mai mare sau egal decat a unui drum. Asadar, raportandu-ne la situatia generala, putem deduce ca pentru orice mers cu circuit, acesta va avea cu siguranta costul mai mare sau egal cu drumul minim pana la nodul prestabilit v , unde $v \in V(G)$.

Prin raportare la concluziile anterioare, un mers de lungime n poate parcurge un circuit C , unde $a(C) = 0$, iar in afara circuitului, acesta reprezinta drumul de cost minim de la nodul sursa s la un nod v , unde $v \in V(D)$. Ceea ce avem de facut in momentul de fata este identificarea unui drum de cost minim P , unde restul impartirii lungimii drumului P la n va da restul $1(\lg(P) \bmod n)$. Conditia este ca drumul sa contina printre elementele sale un nod care se afla pe circuitul C .

Consideram t unul din nodurile care se afla pe circuitul C . Fie $u_{t,t}$ drumul de cost minim pornind din nodul sursa s . Un mers care satisface conditia impusa, va concatena drumul de la nodul sursa s la t , indiferent de cate parcurgeri succesive ale circuitului C exista. Prin urmare, vom adauga un numar k de parcurgeri ale circuitului C incepand din nodul p , astfel incat $l_t + k \cdot l \geq n$, cu k minim. Asadar, vom avea un mers de cost minim intre nodurile s si t . Asociem prefixul acestui mers care va avea ca punct de oprire cel de-al $(n+1)$ -lea element al sau. Fie elementul de pe pozitia $(n+1)$ nodul v , unde $v \in V(D)$. Consideram adevarata afirmatia conform careia un prefix al unui mers de cost minim reprezinta si el un mers de cost minim. Asadar, in cazul de fata, prefixul identificat va avea costul minim asociat drumului dintre s , respectiv v .

Prin mentionarea aspectelor de mai sus, putem conchide ca problema a fost rezolvata, mersul cerut a fost identificat, iar varful v exista proprietatea $A_n(v) = \min\{a(P) | P \text{ drum de la } s \text{ la } v \text{ in } D\}^*$ fiind satisfacuta..

b) Pentru rezolvarea cerintei, consideram $\lg(X)$ -numarul de arce din X , respectiv $a_{avg}(x) = \frac{a(X)}{\lg(X)}$. Pornind de la $(a_{avg})^* = 0$, trebuie sa demonstram ca:

$$(a_{avg})^* = \min_{v \in V} \max_{0 \leq k \leq n-1} \frac{A_n(v) - A_k(v)}{n - k} \quad (1)$$

Pentru a fi indeplinita relatia, luand in considerare valoarea lui $(a_{avg})^*$, deducem ca valoarea raportului va fi egala cu 0, doar in cazul in care $A_n(v) - A_k(v) = 0$, asta tinand cont si de faptul cu numitorul este nenul. Stim ca $k < n$, deci vor exista 2 drumuri de acelasi cost, dar cu lungimi diferite, fapt ce implica ca o parte din muchiile care compun drumul de la nodul de start la v sa aibe asigurate costuri nule.

Deoarece din ipoteza subpunctului **b)** $(a_{avg})^* = 0$ inseamna ca exista un circuit al carui cost mediu este 0. Vom nota cu r -lungimea corespunzatoare drumului de cost minim din nodul de start s , $u_{r,v} = w_{r,v}$ (drumul minim este si mersul minim de lungime k , tinand cont si de algoritmului Bellman-Ford-Moore). Prin urmare, cum pentru alegerea lui v s-ar respecta $w_{n,v} = u_{r,v}$, am avea $w_{n,v} - w_{r,v} = 0$.

Notam faptul ca prin definitie, $u_{r,v} = \min_{0 \leq k \leq n-1} A_k(v)$ (drumul de cost minim este minimul mersurilor de cost minim dupa lungime). Deci $0 = w_{n,v} - w_{r,v} = \max_{0 \leq k \leq n-1} (A_n(v) - A_k(v)) \geq 0$ (deoarece $A_k(v) \geq u_{r,v}$).

Vom considera ca egalitatea va fi indeplinita daca vom alege cazul particular $k = r$, adica lungimea drumul de cost minim va fi k . In acest mod, relatia $(a_{avg})^* = \min_{v \in V} \max_{0 \leq k \leq n-1} \frac{A_n(v) - A_k(v)}{n - k} = 0$ este satisfacuta. **q.e.d.**

c) Stiind ca $(a_{avg})^* \neq 0$, vom modifica functia care asigneaza costul muchiilor in urmatoarea maniera: din costurile vechi ale muchiilor vom scadea $x = (a_{avg})^*$. Deci, noua functia de cost a' va fi egala cu $a - (a_{avg})^*$. Pentru aceasta alegere aduce urmatoarele argumente:

Pentru ca relatia (MMC) sa fie satisfacuta, este necesar ca $(a'_{avg})^* = 0$.

Dar $(a'_{avg})^* = \min_{C \text{ circuit in } D} a'_{avg}(C)$ este costul mediu minim al unui circuit in D . Alegand circuitul de cost minim pentru a , observam ca pentru noua forma propusa pentru functia a , constanta x pe care o scadem din functia de cost a va fi direct proportionala cu $n - k$, rezultand ca termen singular in relatia (MMC):

$$\frac{A'_n(v) - A'_k(v)}{n - k} = \frac{A_n(v) - n \cdot (a_{avg})^* - A_k(v) + k \cdot (a_{avg})^*}{n - k} = \frac{A_n(v) - A_k(v) - (n - k) \cdot (a_{avg})^*}{n - k} = \frac{A_n(v) - A_k(v)}{n - k} - (a_{avg})^*$$

Identificand circuitul de cost minim D , vom calcula noul cost mediu minim $(a'_{avg})^*$. $a(D) = k \cdot a_{avg}$ iar $a'(D) = k \cdot a_{avg} - k \cdot a_{avg} = 0$ unde $a_{avg} = \min_{v \in V} \max_{0 \leq k \leq n-1} \frac{A'_n(v) - A'_k(v)}{n - k} = \min_{v \in V} \max_{0 \leq k \leq n-1} \frac{A_n(v) - A_k(v)}{n - k} -$

$(a_{avg})^*$) acest lucru rezultand si conform subpunctului **b**) unde $(a_{avg})^* = \min_{v \in V} \max_{0 \leq k \leq n-1} \frac{A_n(v) - A_k(v)}{n-k} = 0$ deci si $(a'_{avg})^* = 0$, iar aceasta este conditia necesara pentru satisfacerea relatiei (MMC). Astfel transformarea functiei de cost a in noua functie $a' = a - (a_{avg})^*$, egalitatea (MMC) corespunzatoare functiei de cost a' va avea loc.

Faptul ca (MMC) se va respecta pentru orice functie de cost a este indeplinit deoarece functia de cost a a fost modificata prin scaderea unei valori reale, constante, valoare ce este independenta de variabile adignite circuitelor din digraf sau digrafului insusi (spre exemplu, variabile ca n sau k). Faptul ca se scade o constanta reala nu particularizeaza functia, ci o face sa isi mentina generalitatea.

3 Problema 3

Pentru a demonstra ca problema **P** se reduce in timp polinomial la testarea daca macar unul din grafurile G_{rosu} sau G_{verde} are cuplaj perfect, vom analiza, mai intai, complexitatea pasilor de constructie ale grafurilor G_c unde $c \in \{\text{rosu}, \text{verde}\}$.

Pasul **i**, reprezentativ pentru construirea grafurilor de tipul $H = K_2 \times (G - \{s, t\})$. Orice varf din multimea $G - \{s, t\}$, va fi reprezentat de alte 2 varfuri nou construite, de forma $(rosu, v)$ si $(verde, v)$, varfuri care vor fi incidente. Ne vor rezulta astfel doua grafuri H_1 si H_2 in care $\forall x \in V(H_1), (rosu, x) \in E(H_1)$ si $|V(H_1)| = |G - \{s, t\}| = n - 2$, tinand cont de faptul ca $|V(G)| = n$, si $\forall y \in V(H_2), (verde, y) \in E(H_2)$ si $|V(H_2)| = |G - \{s, t\}| = n - 2$, H_1 si H_2 fiind 2 partitii ale grafului H . H devine astfel un graf bipartit, iar prin faptul ca orice 2 noduri de forma $(rosu, v)$ si $(verde, v)$ sunt incidente, va exista atfel in H un cuplaj perfect, H avand toate nodurile saturate, $S(M) = V(H)$, unde M -cuplaj. Acesta constructie se va realiza in timp polinomial $O(|V(G)|^2) = O(n^2)$ fiind necesara parcurgerea muchiilor grafului.

Urmatorul pas de constructie, **ii**, asigura clasificarea muchiilor din G in cele doua noi subgrafuri H_1 si H_2 , in functie de culoarea ce le-a fost atribuita, dar si mentinerea culorilor din G . Acest pas urmareste ca muchiile uv sa nu aiba 2 culori diferite in acelasi subgraf. Mai concret, daca muchia $uv \in E_{rosu}(G)$, atunci muchia $(rosu, u)(rosu, v)$ va exista in subgraful H_1 ce contine noduri de forma $(rosu, x), x \in V(H_2)$, iar muchia $(verde, u)(verde, v)$ va fi stearsa din subgraful H_1 . Pentru fiecare muchie din G , se verifica atributul ei cromatic si se va sterge din unul dintre subgrafuri H_1 sau H_2 muchia necorespunzatoare caracteristicilor partitiei din care este apartenenta.

Parcurgerile grafului G dar si ale subgrafurilor H_1 si H_2 se pot realiza concomitent doar prin parcurgerea muchiilor grafului G . Acest lucru este posibil prin prisma faptului ca cele doua subgrafuri au fost contruite avand la baza nodurile grafului G , deci printr-o simpla parcurgere a muchiilor grafului G , putem face referinta atat la muchiile grafului initial cat si la muchiile din cele doua subgrafuri fara costuri suplimentare de timp. Grafurile $G - \{s, t\}$, H_1 si H_2 sunt izomorfe, lucru mai mult decat evident, deci cum vizitam 3 grafuri izomorfe simultan, o tranzitie la o muchie noua ar insemna 3 tranzitii in total.

Operatia de sterge are cost unitar de timp, se realizeaza in $O(1)$, deoarece nu sunt necesare modificari suplimentare, fiind doar eliminata muchia necorespunzatoare din multimea muchiilor subgrafului respectiv. Deci si aceasta operatie se rezuma la o parcurgere a tuturor muchiilor proportionala ca timp cu parcurgerea muchiilor grafului G . Deci, prin cele mentionate anterior, timpul total pentru pasul **ii** va fi tot polinomial, $O(|V(G)|^2) = O(n^2)$.

Este de observat similaritatea pasilor **iii** si **iv**, acestia constand in adaugarea muchiilor $s(c, u)$ respectiv $(c, u)t$ care corespund muchiilor din $E(G)$ si corespondentelor lor cromatice. Se vor adauga aceste 2 muchii in asa fel incat drumul de la s la t sa fie un *drum colorat*, adica prima si ultima muchie sunt de aceeasi culoare si orice doua muchii succesive de pe drum sunt de culori diferite. In fiecare din cei doi pasi, se va face verificarea pentru maxim $|V(G)| - 2$ muchii, tinand cont ca $|V(G - \{s, t\})| = n - 2$. Deci in ceea ce priveste timpul de executie acesta va fi tot polinomial, respectiv $O(|V(G)|) = O(n)$.

Tinand cont de argumentele enuntate anterior, am justificat astfel faptul ca toti pasii de constructie se vor realiza in timp polinomial, in cazul cel mai nefavorabil rezultand un timp de $O(n^2)$.

Acum vom demonstra ca problema **(P)** se reduce in timp polinomial la testarea daca macar unul din grafurile G_{rosu} sau G_{verde} are cuplaj perfect.

Din enuntul problemei, un *drum colorat* de la s la t , are ca definitie o succesiune de muchii de la s la t , de culori alternante in care prima si ultima muchie sunt din aceesi clasa a bipartitiei si orice doua muchii succesive de pe drum sunt din clase diferite ale bipartitiei. Cum culorile pot fi simetrice, pentru exemplificare vom alege ca prima muchie ce pleaca din nodul de start S sa aiba culoarea *rosu*.

Existenta unui astfel de drum este echivalenta cu existenta unui drum de crestere in G de la s la t . Se numeste *drum de crestere* al lui G relativ la cuplajul M un drum alternat cu extremitatile varfuri distincte, expuse relativ la cuplajul M . Conform teoremei lui **Berge 1959**, un cuplaj M este de cardinal maxim in graful G daca si numai daca nu exista in G drumuri de crestere relativ la M .

Va exista un drum colorat in G intre s si t daca si numai daca avem un cuplaj perfect in subgraful G_{rosu} . Alegerea arbitrara a primei si a ultimei muchii de culoare rosie, facuta anterior, va determina ca numarul de muchii rosii din drumul de crestere sa fie mai mare cu 1 fata de numarul de muchii verzi.

Pasii **i** si **ii** conduc la obtinerea unui graf ce are $2n$ noduri in care poate fi regasit un cuplaj perfect de marime n dat de muchiile $(rosu, v)(verde, v)$, $\forall v \in V(G)$. Iar pentru urmatorii pasi, **iii** si **iv**, se vor adauga muchiile asociate lui s si lui t avand culoarea corespunzatoare.

Pentru implicatia directa, " \implies ", in G_{rosu} drumul colorat va fi de forma $s(rosu, v_1) \rightarrow (rosu, v_1)(verde, v_1) \rightarrow (verde, v_1)(verde, v_2) \rightarrow \dots \rightarrow (verde, v_k)(rosu, v_k) \rightarrow (rosu, v_k)t$. Se poate observa ca excluzand muchiile exterioare (cele care sunt incidente cu s si t), drumul are un cuplaj perfect sau 1-factor. Adaugand cele 2 muchii eliminate anterior la un cuplaj perfect, sustine faptul ca drumul colorat va deveni astfel un drum de crestere. Nodurile s si t sunt expuse. Prin adaugarea unei muchii la cuplajul obtinut, prin faptul ca acesta era perfect si avem 2 noduri expuse, adaugarea unei muchii va genera un alt cuplaj perfect, de marime $n + 1$. Cuplajul va fi perfect deoarece satureaza toate varfurile grafului G si datorita faptului ca G_{rosu} este de ordin $2n + 2$, va fi exact dublul cuplajului gasit anterior.

Implicatia reciproca, " \impliedby " consta in a presupune existenta unui cuplaj perfect, si vom alege in mod arbitrar G_{rosu} . Consideram deci muchiile incidente in s si t ca fiind acoperite de acest cuplaj, acesta saturand toate nodurile de pe parcursul drumului, s fiind nodul de start si t fiind nodul terminal. Existenta cuplajului perfect este un argument mai mult decat suficient pentru argumentarea existentei drumului colorat, alternanta de cuplat-necuplat ajutand la construirea unui drum alternat al lui G relativ la cuplajul G_{rosu} .

Prin cele enuntate anterior, am demonstrat ca problema **P** se reduce in timp polinomial la testarea daca macar unul din grafurile G_{rosu} sau G_{verde} are cuplaj perfect. **(q.e.d)**

4 Problema 4

Consideram $G=(S, T; E)$ un graf bipartit. Dorim sa demonstram ca graful G are un subgraf partial in care orice varf din S are gradul k ($k \in \mathbb{Z}_{\geq 0}$) si orice nod din T are gradul cel mult 1, daca si numai daca conditia este indeplinita:

$$\bullet (k \cdot H) \quad | N_G(A) | \geq k | A |$$

Pentru a putea demonstra cele enuntate mai sus, vom recurge la utilizarea teoremei lui Hall, care ne va conduce la rezultatul vizat.

Fie G un graf bipartit si S , respectiv T , partițiile sale. Un cuplaj de ordin k in G este o submultime M , unde $M \in E(G)$, astfel incat $\forall v \in S$, v este incident cu exact k muchii din multimea M si $\forall w \in T$, unde w este incident cu maxim o muchie din multimea M . Fie G un graf bipartit

fara noduri izolate. Spunem ca G are un cuplaj de ordinul k care satureaza nodurile din S in T , daca si numai daca satisface conditia lui Hall de ordinul k in S .

Demonstratie:

Implicatia directa: Fie M un cuplaj de ordinul k in graful G . Formam un subgraf R al lui G , stergand toate muchiile din G care nu se afla in M . Facem observatia ca R contine acelasi numar de noduri ca G . Astfel, M satureaza nodurile din S in T si in subgraful nou creat. In subgraful R , este de remarcat ca $\forall v \in V(R)$, v are k vecini distincti in T , iar oricare ar fi 2 noduri din S , ele nu pot avea acelasi vecin in T , prin definitia cuplajului M . Prin urmare, pentru oricare submultime A a lui S , conditia $|N_G(A)| \geq |N_R(A)| = k |A|$ este respectata.

Implicatia inversa: Vom proceda prin inductie dupa numarul de elemente din S . Vom considera cazul de baza: $|S| = 1$. Este usor de observat ca, odata ce G satisface conditia lui Hall de grad k in S , singurul nod existent in S are gradul cel putin k . Luand M o multime de oricare k muchii distincte din graful G , incidente cu singurul nod din S , gasim astfel cuplajul cautat.

Presupunem ca teorema se aplica pentru $|S| < n$. Luam $|S| = n$, unde $n > 1$. Fie G^* un subgraf al lui G , avand aceeasi multime de noduri ca G . De remarcat faptul ca G^* este tot un graf bipartit cu $S^* = S$ si $T^* = T$. Vom demonstra ca G^* are un cuplaj care satisface S^* in T^* si ca acelasi cuplaj va putea fi aplicat si pe graful initial G .

Cazul 1: presupunem ca $\exists e \in E(G^*)$, astfel incat $\exists A^* \subsetneq S^*$ si $|N_{G^*-e}(A^*)| < k |A^*|$. Avand deja $|N_{G^*}(A^*)| \geq k |A^*|$, inseamna ca $|N_{G^*}(A^*)| = k |A^*|$. Acum vom crea doua subgrafuri ale lui G^* . Vom defini un graf bipartit G_1^* ale carui partitii sunt $S_1^* = A^*$, respectiv $T_1^* = N_G(A^*)$. Definim un alt graf bipartit G_2^* ale carui partitii sunt $S_2^* = S^* - S_1^* = S^* - A^*$ si $T_2^* = T^* - T_1^*$. Doua varfuri din $V(G_1^*)$ sunt adiacente in G_1^* , daca si numai daca aceste doua varfuri sunt adiacente in G^* . Similar pentru G_2^* . Este de remarcat ca G_1^* satisface conditia lui Hall de ordin k in S_1^* . Deci, presupunem ca $|S_1^*| < |S^*| = n$. Deci, prin inductie, exista un cuplaj M_1^* de ordin k , care satureaza S_1^* in T_1^* in graful G_1^* . Acum vom arata ca si in G_2^* este satisfacuta conditia lui Hall de ordin k . Fie $Q^* \subseteq S_2^*$. Stim ca $|N_{G^*}(Q^* \cup A^*)| \geq k |Q^*| + k |A^*|$. Daca $|N_{G_2^*}(Q^*)| < k |Q^*|$, atunci am avea $|N_{G^*}(Q^* \cup A^*)| < k |Q^*| + k |A^*|$, fapt ce determina o contradictie. Deci avem ca $|N_{G_2^*}(Q^*)| \geq k |Q^*|$, $\forall Q^* \subseteq S_2^*$. Acum, cat timp avem $|X_2^*| < |X^*| = n$, atunci, prin inductie, vom avea un cuplaj M_2^* de grad k , care va satura S_2^* in T_2^* in graful G_2^* . Este evident ca $M_1^* \cup M_2^*$ va fi considerat cuplajul cautat pentru G^* , asadar si pentru G .

Cazul 2: presupunem ca $\exists e \in E(G^*)$, astfel incat $\exists A^* \subsetneq S^* \Rightarrow |N_{G^*-e}(A^*)| \geq k |A^*|$. Fie $v \in S^*$ si w_1, w_2, \dots, w_k cei k vecini distincti ai nodului v . Fie $e_1 = vw_1 \in E(G^*)$, Fie $j \in \{1, \dots, k\}$. Stim ca $G^* - e_j$ nu satisface conditia lui Hall in S^* . Dar in acest, caz, singura posibilitate $|N_{G^*-e_j}(S^*)| < k |S^*|$. Daca ar fi fost asa incat $w_j \in N_G(S^* - v)$, atunci am fi avut $N_{G^*-e_j} = N_G(S^*)$. Dar tinand cont ca $|N_G(S^*)| \geq k |S^*|$ am fi avut $|N_{G^*-e_j}(S^*)| \geq k |S^*|$, fapt ce conduce la o contradictie. Prin urmare $w_j \notin N_G(S^* - v)$, $\forall j \in \{1, \dots, k\}$.

Scriem $M^* = \{vw_i \in E(G^*) : i = 1, \dots, k\}$. Definim un graf bipartit G_0^* , care satisface conditia lui Hall de grad k pe S_0^* . Deci, tinand cont ca $|S_0^*| < |S^*| = n$, prin inductie exista un cuplaj M_0^* de ordin k , care satureaza S_0^* in T_0^* in G_0^* . Este usor de remarcat ca $M^* \cup M_0^*$ reprezinta cuplajul cautat pentru G^* si totodata pentru graful initial G . Luand in considerare cazurile prezentate, demonstratia este completa.