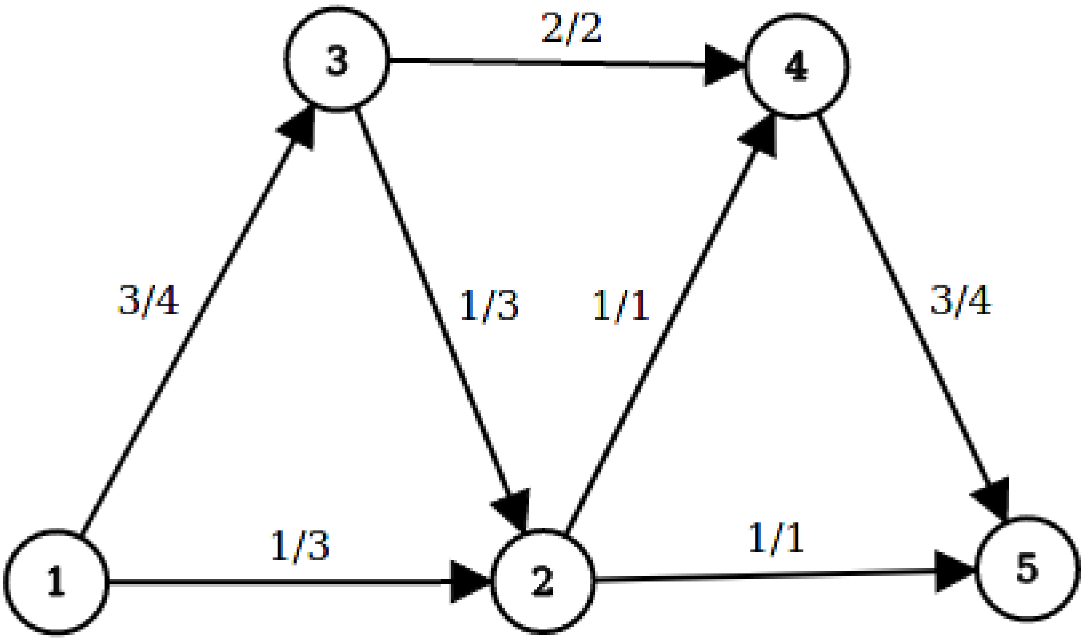
Tema 2 - Algoritmi Fundamentali

Nicoleta Ciaușu

January 15, 2021

Exercitiul 1

Fluxul maxim din S in T este 4. Dupa o rulare de algoritm de determinare a fluxului maxim (Ford-Fulkerson) graful poate arata asa:

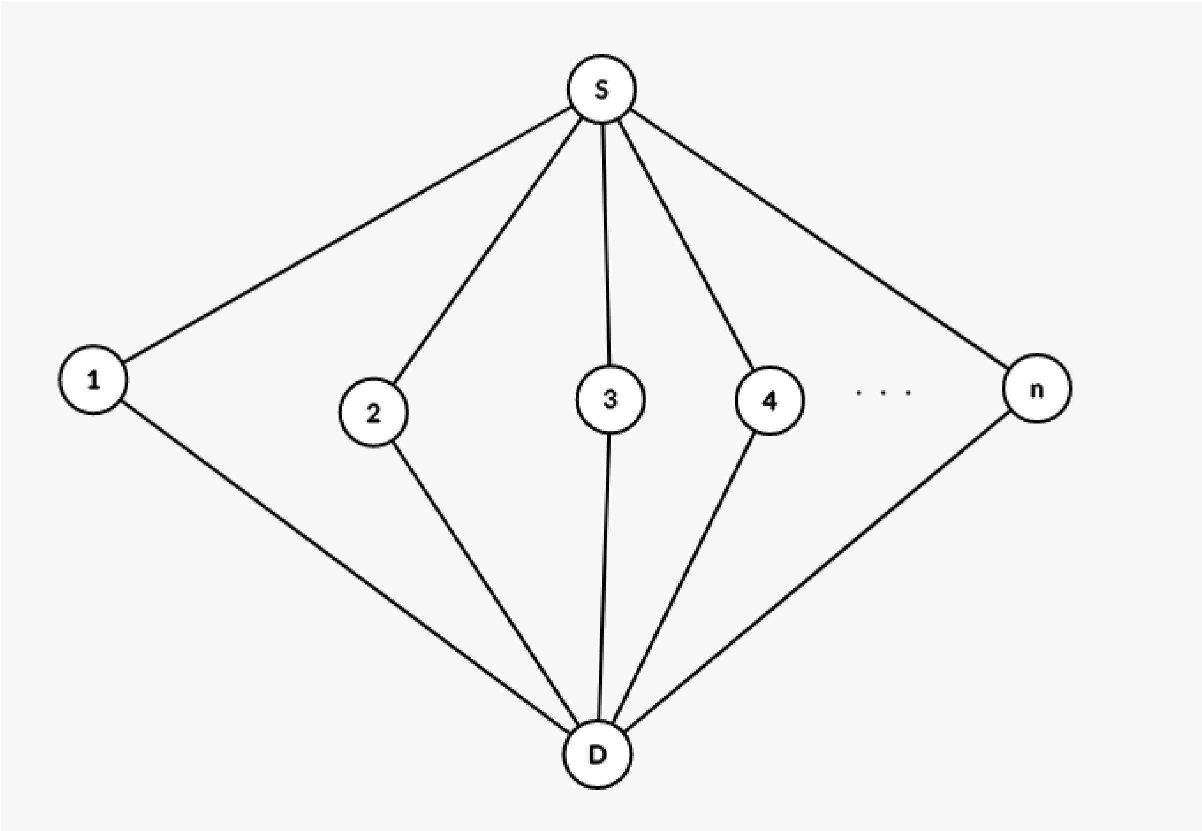


Iar taietura de cost minim este data de muchiile saturate, in acest caz $3 \longrightarrow 4, 2 \longrightarrow 4, 2 \longrightarrow 5$.

Exercitiul 2

Cum Edmonds-Karp foloseste BFS pentru a gasi lanturi de augumentare, si ii putem asocia complexitatea $\Omega(E)$, caci trebuie sa parcurga fiecare muchie, ar trebui sa gasesc o retea de flux pe care BFS sa ruleze de $\Omega(E)$ ori.

Fie urmatoarea retea care respecta aceasta cerinta:



Parcurgerea acestei retele necesita $\Omega(E/2)$ iteratii de BFS, deoarece se va face o augumentare pentru fiecare nod, si exista dublu numar de muchii.

Cum $\Omega(E/2) \approx \Omega(E)$, rezulta o complexitate de $\Omega(E^2)$ pe acest graf.

Exercitiul 3

Exercitiul 4

Voi incerca sa transform constrangerea in felul urmator: pentru fiecare nod X de limita C voi crea un nod X' si voi redistribui muchiile adiacente lui X astfel:

- Muchiile care intra in X vor ramane in X
- Muchiile care ies din X vor iesi acum din X'
- Intre X si X' voi pune o muchie cu cost C

Pe graful rezultat voi putea aplica orice algoritm de flux maxim iar complexitatea va ramane aceeasi deoarece se modifica doar numarul de muchii/noduri prin constante (se dubleaza numarul de noduri si se adauga inca n muchii), nefiind necesara modificarea algoritmului in sine.

Exercitiul 5

I. Vreau sa configurez arborele T astfel incat sa pot aplica un algoritm de flux maxim pe el. Lui T ii voi face urmatoarele modificari:

- Voi crea un nod nou S pe care il voi lega printr-o muchie de toate orasele. Capacitatea muchiei legate la nodul i va fi s_i.
- Voi crea un nod nou D si voi conecta printr-o muchie fiecare oras i la el cu capacitatea d_i;
- Intre doua orase putand sa circule oricati oameni, muchia care conecteaza orasul i de parintele sau va avea capacitate infinita.

Voi considera fiecare cetatean care isi doreste sa isi gaseasca o casa drept *o unitate de flux*. Un cetatean poate alege sa ramana pe loc intr-un oras sau sa urce in arbore in cautarea unui alt loc. Operatiunea de *stabilire* intr-un oras presupune trimiterea unitatii de flux in D.

La sfarsitul rularii algoritmului de flux maxim, vom avea o configuratie valida daca si numai daca fluxul maxim este egal cu numarul de oameni.

II. Data fiind structura arborescenta a grafului, putem rezolva problema in $O(N)$ aplicand la fiecare o operatie de comprimare a frunzelor.

Fiecarui nod i ii corespund 3 muchii: $S \longrightarrow I, I \longrightarrow D, I \longrightarrow parinte$. Saturand muchia $I \longrightarrow D$ cu fluxul care vine din S, orice alt om care va mai ajunge in orasul I va alege sa urce spre parintele sau. Asadar, cum practic nu mai este nevoie de nodul I, il voi sterge si voi duce muchie din S in D. Capacitatea acestei muchii va fi data de diferenta dintre d_i si s_i.

- In cazul in care $s_i > d_i$, voi suplimenta $s_{parinte[i]}$ cu $s_i - d_i$ iar capacitatea muchiei noi de la S la D va fi d_i.
- Daca $s_i < d_i$, capacitatea muchiei noi de la S la D va fi s_i .

Cum atat operatia de comprimare, cat si cea de determinare a fluxului au complexitate $O(1)$, si avem N noduri, rezulta un algorim de complexitate $O(N)$.

Exercitiul 6

Fie G graful posibilitatilor de transferare. Voi incerca sa modelez acest graf intr-o retea de flux sub forma de graf bipartit cu N noduri in stanga si N noduri in dreapta, unde N este numarul echipelor.

- Fie doua noduri S si D. Voi conecta S la toate nodurile din stanga, respectiv toate nodurile din dreapta la D, cu muchii de cost 0 si capacitate 1. Astfel, se va face un singur transfer de la o echipa la alta (prima conditie a problemei).
- Intre doua echipe intre care se poate efectua un transfer voi duce o muchie de cost $-pretTransfer$ si capacitate 1.
- Voi mai duce muchie de la nodul X_i *din stanga* la nodul X_i *din dreapta* cu cost 0 pentru a modela alegerea de a nu face niciun transfer cu echipa X_i.

In final, raspunsul va fi $-1 \cdot c$, unde c este fluxul maxim de cost minim din reteaua modelata mai sus.

In plus, putem fi siguri ca fiecare echipa va avea acelasi numar de jucatori la final deoarece transferarea dintr-o echipa a in b satureaza muchia $b \longrightarrow D$ ceea ce inseamna ca nu se mai poate face transferul nici de la alta echipa in b, nici de la b in b (ceea ce inseamna ca echipa b este si ea obligata acum sa transfere un jucator altei echipe).

Exercitiul 7

I. Fie G graful preferintelor dansatorilor. Cum perechile sunt de forma *baiat* $\iff$ *fata* vom avea un graf bipartit intre baieti si fete. Vom elimina din graf toate muchiile care nu sunt duble (ne intereseaza doar optiunile valide, acelea in care atat baiatul b_i cat si fata f_i se au reciproc pe lista de optiuni).

Fiecarei muchii $b_i \longrightarrow f_i$ ii voi asocia capacitate 1. Voi crea o retea de flux legand un nod S de catre toti baietii si voi lega fiecare fata la un nod D. Capacitatea muchiilor din $S \longrightarrow b_i$ si $f_i \longrightarrow D$ va fi K (numarul de runde), deoarece atat fiecare baiat cat si fiecare fata trebuie sa danseze de K ori.

Daca in urma rularii algoritmului de flux obtinem fluxul maxim $N \cdot K$ inseamna ca exista o configuratie valida.

II. Pentru a reconstitui coregrafia vom recrea graful bipartit al preferintelor dansatorilor, cu muchiile inutile eliminate, si vom rula algoritmul de cuplaj de K ori si afisa perechile rezultate, cu mentiunea ca dupa fiecare rulare a algoritmului voi avea grija sa elimin muchiile folosite (deoarece o muchie poate fi folosita o singura data per coregrafie).