Tema 2 Algoritmi Fundamentali

Buhai Darius

January 14, 2021

Exercitiul 1

Fie fluxul:

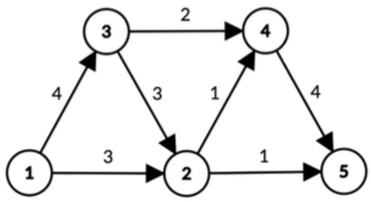


Figure 1: Flux

Fluxul $S - T$ de capacitate maximă pe rețeaua dată, unde $S = 1$ și $T = 5$ este **4**. O $S - T$ tăietură minimă a fluxului care ne indică capacitatea maximă de **4**, este descrisă în figura de mai jos:

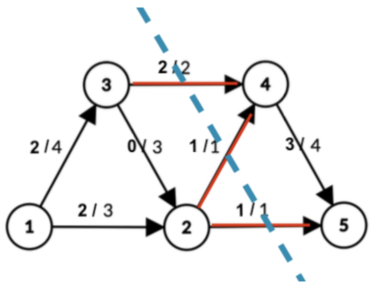


Figure 2: Tăietură minimă

Exercitiul 2

Pentru a demonstra complexitatea algorimului Edmonds-Karp, vom determina pentru început numărul de iterații de BFS ce trebuie rulate.

În rețeaua de flux definită mai jos, algoritmul nostru va rula exact $V \cdot \frac{1}{2} - 2$ interații de BFS, unde $E = O(V)$, deci complexitatea inferioară va fi $\Omega(E)$.

Evident, algoritm-ul BFS rulează în $\Omega(E + V)$ iar în consecință, complexitatea totală a algoritmului Edmonds Karp va fi $\Omega(E^2)$.

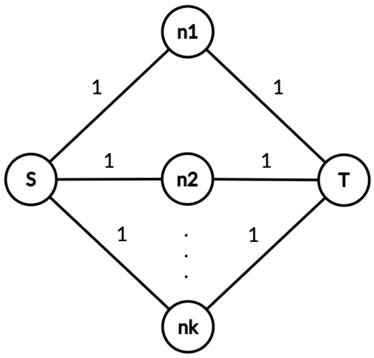
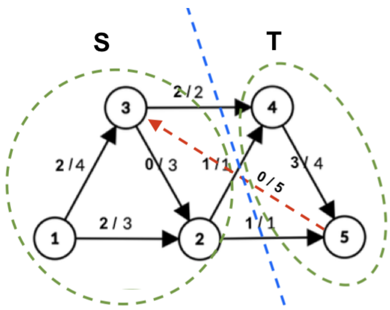


Figure 3: BFS worst case

Exercitiul 3

Demonstrați că în orice rețea de flux, fluxul maxim coincide cu fluxul maxim în subrețeaua care nu conține muchii care pornesc din T sau ajung în S



Fie fluxul de mai sus cu o tăietură minimă ce separă rețeaua în 2 mulțimi de noduri ce îl conțin pe S și pe T.

Conform algoritmului de determinare a tăieturii minime, în urma determinării fluxului maxim, pe graful residual muchiile care pornesc din mulțimea S și ajung în mulțimea T sunt saturate.

Cu toate acestea, dacă o muchie care pornește din T sau ajunge în S trimite flux, atunci în graful residual trebuie să existe și o muchie inversă (din S în T) nesaturată, ceea ce va rezulta într-o contradicție.

Astfel, fluxul maxim coincide întotdeauna cu fluxul maxim în subrețeaua care nu conține muchii ce pornesc din T sau ajung în S.

În aceeași oridine de idei, pe muchiile care pornesc din S și ajung în T nu se va întoarce flux niciodată, deoarece muchiile din tăietura minimă vor fi deja saturate.

Exercitiul 4

Să considerăm o variantă generalizată a fluxului maxim, în care avem constrângerea, în plus, că fiecare nod $v (v \in V)$ are o limită superioară $c(v)$ pentru valoarea fluxului care se transferă prin acel nod.

Pentru a rezolva problema, putem transforma constrângerea la nivel de noduri într-una la nivel de muchii. Astfel, pentru fiecare nod v vom adăuga un alt nod u și o muchie direcțională între cele 2 noduri cu o capacitate $c(u, v) = c(v)$, unde $c(v)$ reprezintă constrângerea la nivel de nod.

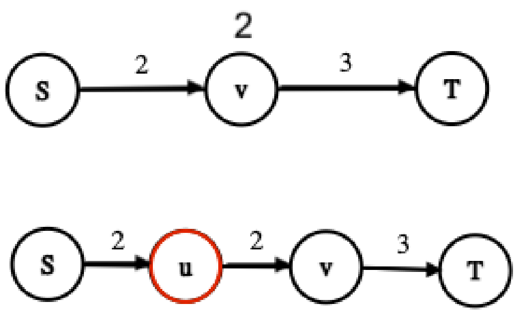


Figure 4: Transformare rețea

În final, putem elimina constrângerile la nivel de noduri și putem calcula fluxul maxim folosind algoritmul Edmonds-Karp ce are o complexitate $\Omega(V^2)$.

Exercitiul 5

Fie T un arbore cu n noduri. Acest arbore descrie harta unei țări; mai exact, putem considera că fiecare nod este un oraș, iar capitala este în orașul 1. În fiecare oraș $i(1 \leq i \leq n)$ există s_i oameni și d_i locuințe ($1 \leq s_i, d_i \leq 10^9$). Un om aflat în oraș v poate să rămână în orașul său sau să se relocheze în orice alt oraș aflat pe drumul dintr v și capitală.

Pentru a modela problema ca una de fluxuri, vom dubla numărul de noduri și vom adăuga un nod sursă (S) și unul destinație (T).

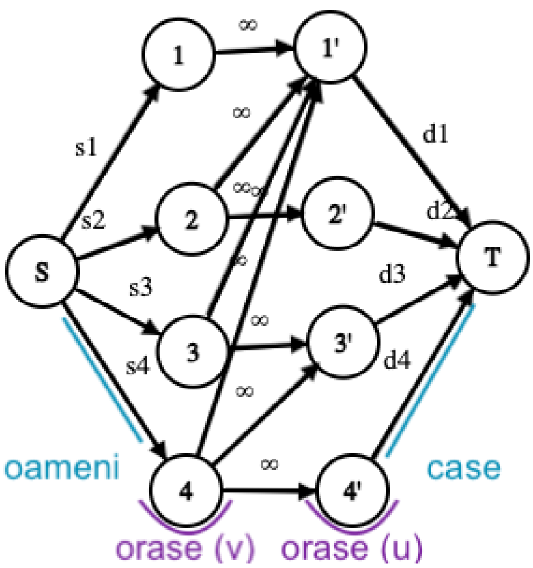


Figure 5: Modelare rețea

Sursa (S) este legată de fiecare din nodurile din mulțimea stângă cu o capacitate s_i, ce reprezintă numărul de oameni din fiecare oraș. De asemenea, vom lega nodurile din mulțimea dreaptă de destinație (T) cu muchii de cost d_i, ce reprezintă numărul de case din fiecare oraș.

Fiecare nod v din mulțimea stângă poate trimite flux în nodurile u din mulțimea dreaptă doar dacă u se află pe drumul dintre v și capitală în arborele inițial. Capacitatea muchiilor intermediare va fi infinită, deoarece nu există nicio limită impusă de oameni ce pot fi mutați.

În final, putem rezolva problema ca una de fluxuri, căutând fluxul maxim ce poate ajunge din S în T, graful nostru rezidual reprezentând strategia de relocare a cetățenilor în orașe.

De asemenea, putem observa cum tăietura minimă pe graful creat poate apărea doar între S și v sau între u și T, simplificând problema la găsirea costului maxim între S și T.

Astfel, o rezolvare eficientă a problemei se poate face folosind metoda greedy, care pornește din fiecare frunză și recursiv pastrează exact atâția oameni câte case are și trimite diferența de oameni mai departe pe drumul către capitală. O astfel de implementare va avea o complexitate $O(N)$ iar rezultatul va fi echivalent cu tăietura minimă a fluxului inițial creat.

Exercitiul 6

Problema dată poate fi modelată ca una de flux maxim de cost minim astfel:

1. Se dublează numărul de noduri din graf.
2. Se adaugă un nod de început (S) și unul de final (T)
3. Nodul de start (S) se leagă de fiecare dintre nodurile din partea stângă a grafului cu muchii cost 0 și capacitate 1
4. La fel, nodul de final se leagă de fiecare dintre nodurile din partea dreată cu muchii de cost 0 și capacitate 1
5. Nodurile din mulțimea stângă pot trimite flux atât catre ele însăși (atunci când echipele nu fac niciun transfer), prin muchii de cost 0 și capacitate 1, cât și către echipele la care vor să facă transferuri, prin muchii de cost $-p_i$ și capacitate 1.

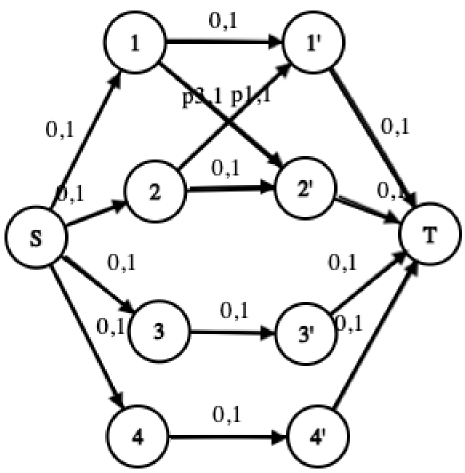


Figure 6: Modelare flux

Rețeaua creată va limita fiecare echipă la maxim un transfer de jucător (muchiile direcționale de legătură au capacitatea 1), iar numărul de jucători din fiecare echipă va rămâne neschimbat în urma transferurilor pentru a păstra un flux maxim egal tăietura minimă **n**.

În final, pentru a rezolva problema eficient, putem folosi algoritmul Ford-Fulkerson optimizat pentru fluxuri maxime de cost minim. Costul maxim rezultat din transferuri va fi egal cu **-min_cost**.

Exercitiul 7

Problema dată poate fi modelată ca un graf bipartit, unde muchiile dintre mulțimile de băieți și fete reprezintă preferințele mutuale cu care fiecare băiat sau fată vrea să danseze.

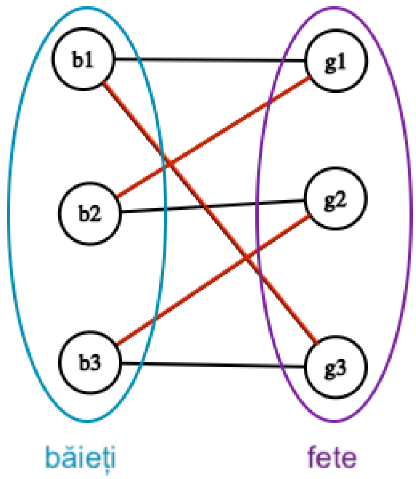


Figure 7: Modelare graf bipartit

Lema Mariajelor menționează că în $\forall G$ graf bipartit, cu mulțimile X (băieți) și Y (fete), $\exists$ un cuplaj perfect între X și Y dacă și numai dacă pentru $\forall$ mulțime W $\in$ X, $|W| \leq |N(W)|$. Evident, dacă fiecare nod din G are gradul egal cu 1, atunci pentru $\forall$ W $\in$ X, $|W| \leq |N(W)|$, ceea ce demonstrează cuplajul perfect.

Din demonstrația anterioară am stabilit că pentru fiecare rundă avem nevoie ca fiecare nod din G să aibă gradul 1, pentru a exista un cuplaj perfect. În continuare, știm că în coregrafia întregului spectacol vor fi realizate k runde, în care, după stabilirea configurației de cupluri, muchiile dintre cele 2 mulțimi de fete și băieți vor fi marcate (eliminate), micșorând gradul nodurilor din G cu 1.

Astfel, din cele 2 afirmații, putem reduce determinarea existenței cuplajului perfect pentru toate cele k runde, la verificarea dacă fiecare nod din graful G are gradul egal cu k, adică dacă graful G este $k - regulat$.

În final, putem găsi configurațiile pentru fiecare din cele k runde folosind algoritmul de cuplaj maxim în graf bipartit, ce are o complexitate totală $O(N \cdot M)$.