Datorii
Autor: Pătcaş Csaba

Împrumuturile făcute definesc un graf orientat. O primă idee ar fi să reducem acest graf în aşa fel încât să aibă număr minim de muchii. Două operaţii de transformare posibile sunt următoarele:

1. Scadem din toate muchiile unui ciclu o valoare fixă.
2. Scadem din toate muchiile unui lanţ o valoare fixă şi introducem o nouă muchie având costul valorii fixe pornind de la începutul lanţului şi ajungând în nodul terminal.

Din păcate contează ordinea în care facem aceste transformări şi se poate întâmpla să ne blocăm într-o stare neoptimă. Autorul nu a reuşit să determine o regulă, care defineşte ordinea corectă a acestor transformări, în aşa fel ca ele să ducă la soluţia optimă.

Abordarea corectă se bazează pe construirea unei reţele de flux. Pentru fiecare persoană vom calcula dacă trebuie să dea bani sau să primească şi vom construi un graf bipartit după acest criteriu. Să numerotăm de la 1 la n1 persoanele care trebuie să dea bani şi de la n1+1 la n1+n2 persoanele care trebuie să primească. Reţeaua de flux se construieşte în modul următor:

1. Adaugăm nodul sursă S şi nodul destinaţie D.
2. De la nodul S avem o muchie la fiecare dintre nodurile 1..n1 având capacitatea egală cu suma de bani pe care persoana respectivă trebuie să o dea.
3. De la fiecare dintre nodurile n1+1..n1+n2 avem o muchie în nodul D, având capacitatea egală cu suma de bani pe care persoana respectivă trebuie să primească.
4. Între fiecare nod dintre 1 şi n1 şi fiecare nod dintre n1+1 şi n1+n2 avem o muchie cu capacitatea infinit.

Problema se reduce la găsirea fluxului maxim folosind număr minim de muchii în acest graf. Abordarea cu flux maxim de cost minim nu funcţionează, pentru că costul se calculează pe unitatea de flux, iar noi putem avea fluxuri mai mari decât 1. În problema noastră avem costuri fixe pe muchiile prin care trece flux strict mai mare ca 0. Această problemă se numeste în literatură Minimum Edge-Cost Flow si este NP. Chiar dacă avem de rezolvat un caz particular al problemei (graf bipartit, costuri egale cu 1), autorul n-a găsit un algoritm polinomial pentru rezolvare.

Limita mică pentru N face posibilă abordarea prin metoda programării dinamice. Vom defini o subproblemă prin starea i,j cu i=0..$2^{n1} – 1$ si j=0..$2^{n2} – 1$ cu semnificatia: numărul minim de muchii prin care se poate satisface fluxul luând în considerare doar nodurile determinate de bitii lui i si j. Pentru rezolvarea subproblemei vom alege toate combinatiile de i' si j', pentru care i AND i' = i', j AND j' = j' si cantitatea de flux care intră în nodurile din submultimea determinată de i' este egală cu cantitatea de flux ce iese din nodurile din submultimea determinată de j'. Valoarea i,j se va determina astfel:

DP i,j = min (DP i XOR i' , j XOR j' + bitcount[i'] + bitcount[j'] – 1), unde bitcount[x] semnifică numărul de biti egali cu 1 din reprezentarea binară a lui x, iar I' si j' au semnificatiile de mai sus.

Relatia de mai sus se bazează pe următoarea observatie: o retea cu n1+n2 noduri construit ca mai sus se poate "rezolva" cu cel mult n1+n2-1 muchii. Dacă există două valori i' si j' care respectă cerintele de mai sus, nodurile determinate de i' si j' vor forma o componentă conexă cu muchiile folosite pentru transportarea fluxului între nodurile respective. Dacă nu există asemenea valori, tot graful trebuie să fie conex, ceea ce se poate obtine în cel mai bun caz cu un număr de muchii egal cu numărul de noduri minus 1, care în cazul nostru este egal cu n1+n2-1.
Această abordare vizitează 2^n stări, dar are complexitatea O(3^n), din cauza necesitătii determinării valorilor i' si j'. În functie de eficienta implementării o asemenea abordare ar trebui să obtine între 60 si 100 de puncte.
Abordări bazate pe algoritmi tip greedy sau backtracking ar trebui să obtină între 30 si 60 de puncte.