Q0: De ce folosim algoritmi aproximativi?

- Problemele din NP-hard (cel putin la fel de dificile ca cele din NP-C) nu au algoritmi fezabili (in timp polinomial) pt determinarea optimului. Asa ca una dintre solutiile de compromis ar fi sa gasim o solutie “aproape” optima.

Q1: ce este factorul de aproximare pentru un algoritm?
Fie ALG - solutia noastra, OPT - solutia optima

In cazul unei probleme de minim, o constanta *c* (supraunitara) se numeste factor de aproximare daca:

$OPT \le ALG \le c*OPT$

In cazul unei probleme de maxim, o constanta *c* (subunitara) se numeste factor de aproximare daca:

$OPT \ge ALG \ge c*OPT$.

Q1.1 In cazul unei probleme de minim: Un algoritm 2-aproximativ poate fi numit 3-aproximativ?

Da - deoarece daca am un algoritm c-aproximativ si un c’>c vom avea

$ALG \le c*OPT \le c'*OPT$ - deci $ALG \le c'*OPT$

Q1.2 Cum putem sa justidicam ca un *c* gasit este tight bound?

Tb sa arat ca daca, in cazul problemelor de minim, iau un c’<c (sau c’>c in cazul celor de maxim) atunci algoritmul meu nu are cum sa fie c’ aproximativ.

Mai simplu este sa gasesc o intrare I pentru care ALG(I) = c*OPT(I)

Probleme:

1. Avem următorul scenariu: Avem $n$ colete de transportat, fiecare avand greutatea de $w_1, w_2, ..., w_n$. Pentru a le transporta, putem folosi un număr de camioane, fiecare avand capacitatea de transport $G$. Presupunem că $w_i \le G$, pentru orice $i$. Ne dorim sa minimizăm numărul de camioane folosite. Considerăm următorul plan de încărcare a camioanelor:

Odată ce avem la dispoziție un camion pt a fi încărcat, iterăm prin mulțimea coletelor, incărcându-le in camion, până când dăm peste primul colet ce nu mai incape. În acel moment considerat că am terminat de încărcat camionul curent și trecem la următorul camion, prima dată încărcând coletul care nu a mai încăput în cel precedent.

a) Arătați, printr-un exemplu simplu, că metoda de mai sus nu furnizează soluția optimă.
b) Arătați totuși că soluția de mai sus este un algoritm 2-aproximativ pentru problema noastră.

Raspuns:

a) G=2 W={1,2,1}
   Alg = (1)(2)(1) OPT=(1,1)(2)
b) Observatie: fiecare pareche de camioane (consecutive) din algoritm transporta o cantitate de colete >G.
   Observatie 2: in medie camioanele o sa fie plinie macar pe jumatate. (?) -vine fix din obs 1: un camion e incarcat pana la W<G; Primul obiect care nu incape, are greutate Wi, va fi incarcat in camionul urmator. Deci suma greutatilor celor 2 camioane >G (in medie fiecare >G/2)
   $OPT >= 1/G * \sum(w_i)$

Cazul 1 (si cel dificil) algoritmul nostru foloseste 2q+1 camioane

pt primele q perechi de camioane vom avea un transport >G

$\sum(w_i)=W>q*G$

W/q>G - deci solutia optima are nevoie de cel putin q+1 camioane

solutia mea ofer 2q+1 camioane < 2*(q+1)<=2*OPT

Cazul 2 (simplu) algoritmul foloseste exact 2q camioane ….

2) Dat fiind algoritmul Load-Balance (Cursul 2, slide 19) să se stabilească dacă următoarea afirmație este adevărată sau falsă.

”Pentru orice instanța a problemei de Load-Balace, exista o anumită ordine a procesării activităților astfel încât algoritmul de tip greedy să dea o soluție optimă”

Dacă afirmația este adevărată, oferiți o demonstrație, altfel, găsiți un contraexemplu.

Da. Ne uitam la solutia optima, apoi “rulam” algoritmul nostru pas cu pas, vedem care este masina *i* care urmeaza sa primeasca o activitate si “avem grija” ca acea activitate sa fie “next in line”.

3) Fie Problema Load Balance, dar cu următoarea modificare: Avem *n* joburi si *m* mașini, doar că pentru primele *k* mașini timpul de lucru al unei activități este înjumătățit. Să se găsească un algoritm bazat pe tehnica greedy care furnizeaza o soluție de cel mult 3xOPT.

Raspuns:

Luam activitatile intr-o ordine oarecare. Fiecare activitate o asociez unei masini astfel incat ea sa se termine cat mai devreme.

Fie $t_{max}$ activitatea de cost maxim.

$OPT>=t_{max}/2$

$$OPT \geq \frac{1}{2 \cdot k + (m-k)} \sum_j t_j$$

Fie J(i) - multimea de activitati asignate masinii i

fie q - masina care, la sfarsitul algoritmului, re incarcatura cea mai mare (load(q) este maxim)

fie p - activitatea adaugata la masina q, astfel incat se obtine load-ul maxim.

load’(q) - incarcatura masinii q inainte de a ii fi atribuita activitatea p

$$ALG = load(q) = load'(q) + \begin{cases} daca\ q \leq k: \frac{t_p}{2} \\ altfel: t_p \end{cases}$$

$$ALG \leq load'(q) + t_p \leq load'(q) + t_{max} = load'(q) + 2 \cdot \left(t_{max}/2\right) \leq load'(q) + 2 \cdot OPT$$

Ramane de demonstrat ca OPT>+ load’(q)