LUCAN CRISTIAN

LOAD BALANCE

2. ALG_1 2-aprox $\Rightarrow$ $ALG_1 \leq 2OPT$ (minimizare)

ALG_2 4-aprox $\Rightarrow$ $ALG_2 \leq 4OPT$ (minimizare)

a) ~~ota~~ $2OPT \leq 2ALG_1 \leq 4OPT$ (avem $\exists$ pt care)

$OPT \leq ALG_2 \leq 4OPT$ ~~(maiavem)~~ $2OPT(J) \leq 2ALG_1(J) \leq ALG_2(J)$

$\Rightarrow$ $(\forall)$ J pt care $ALG_2(J) \geq 2OPT$ ~~și $ALG(J)$~~ este valid

~~b)~~

b) $2OPT \leq 2ALG_2 \leq 8OPT$

$OPT \leq ALG_1 \leq 2OPT$

pentru $ALG_1(J) \geq 2ALG_2(J)$, îl putem alege doar pe J care satisface $ALG_1(J) = 2OPT$ și $ALG_2(J) = OPT$

3. t_g g $\rightarrow$ ultima activitatea adăugată la mașina cu loadul maxim (cel care dă răspunsul la algoritm)

~~k~~ $\rightarrow$ mașina la care se adaugă g

load'(~~k~~) $\rightarrow$ loadul mașinii i înainte ca activitatea g să fie asociată mașinii K

$ALG = load'(K) + t_g$

~~$load'(K) < \frac{1}{m} \sum_{1 \leq i \leq g}$~~ ne asumăm că $t_g \leq t_m$ (g este pus după distribuirea primelor m activități), o să iau și cazul celălalt după ...

$load'(K) \leq \frac{1}{m} \sum_{1 \leq i \leq g} t_i < \frac{1}{m} \sum_{1 \leq i \leq n} t_i - \frac{1}{m} t_g$ (1)

$t_g \leq t_m + t_{m+1} \leq t_m + t_m \leq OPT$

$(t_m \geq t_{m+1})$

~~$t_g \leq t_m \Rightarrow 2t_g \leq 2t_m \leq 2OPT \Rightarrow t_g \leq$~~

~~(asumptie)~~

$t_g \leq t_m \Rightarrow 2t_g \leq 2t_m \leq OPT \Rightarrow t_g \leq \frac{1}{2} OPT$ (2)

Din (1), (2) $\Rightarrow load'(K) < OPT - \frac{1}{2m} OPT$

$$\frac{t_g \leq \frac{1}{2} OPT \qquad (+)}{ALG \leq \frac{3}{2} \cancel{OPT} - \frac{1}{2m} \quad OPT\left(\frac{3}{2} - \frac{1}{2m}\right)}$$

TRAVELLING SALESMAN PROBLEM

1. a) Fie A mulțimea grafurilor ce au doar muchii de cost 1 sau 2

Presupunem prin absurd că TSP nu este NP-hard pentru grafurile din mulțimea A.

Fie $G \in A$, $(\forall)\ e \in$ muchii, $cost(e) = 1 \Rightarrow (\forall)$ ciclu c hamiltonian are cost n, unde n = nr de noduri

Completăm G, ~~astfel~~ muchiile nou adăugate având cost 2 (notăm acest graf complet G')

Deci pentru $e \in G \Rightarrow cost_{G'}(e) = 1$

$e \notin G \Rightarrow cost_{G'}(e) = 2$

Rulând TSP pe G' putem obține 2 tipuri de output:

output $= n \Leftrightarrow (\exists)\ c$ ciclu hamiltonian în G (am folosit doar muchii din G, de cost 1)

output $> n \Leftrightarrow (\nexists)\ c$ ciclu hamiltonian în G (am folosit măcar o muchie de cost 2, adăugată în G')

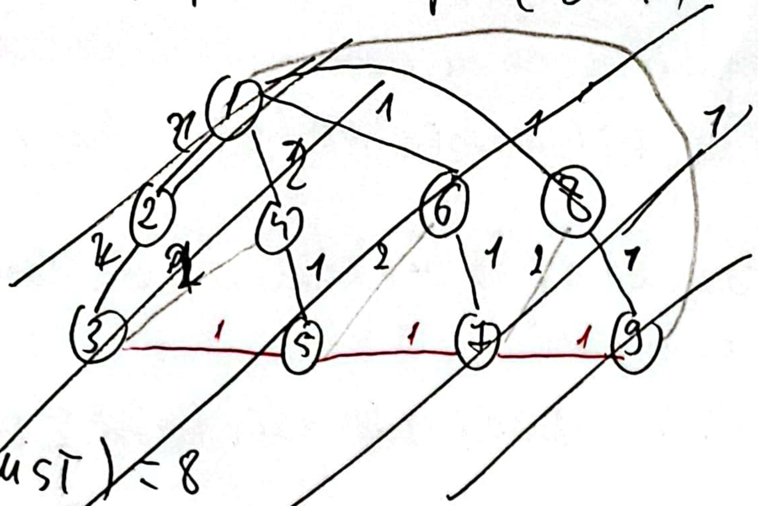
Asta înseamnă că rulând TSP pe G' putem decide dacă $(\exists)\ c$, ciclu hamiltonian în G.

~~Determinarea unui ciclu hamiltonian în G este NPC (deci și NP-hard)~~

TSP pe G' ~~este~~ nu este NP-hard (ipoteză) $\Leftrightarrow$ HCP în G nu este NP-hard (fals)

$\Rightarrow$ TSP este NP-hard pentru grafuri din A

c) Fie G graful de mai jos (G ∈ A)



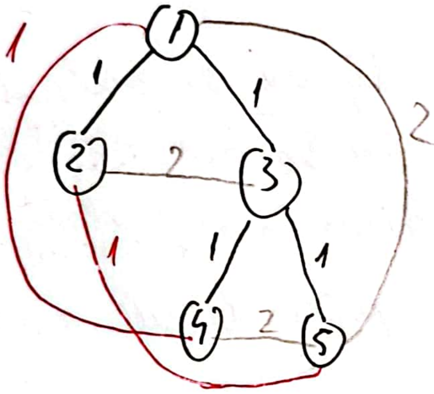
— muchii MST

— muchiile folosite de algoritmul aproximativ pe lângă cele din MST

— ~~muchiile alese de O~~ muchiile alese de OPT ~~Ab~~ în locul celor alese de ALG (cele cu —)

Cost(MST) = 8

Cost(ALG) = 12



Cost(MST) = 4

Cost(ALG) = 8 (1, 2, 3, 4, 5, 1)

Cost(OPT) = 5 (4, 3, 5, 2, 1, 4)

$\frac{Cost(ALG)}{Cost(OPT)} = 1,6 > \frac{3}{2} \Rightarrow$

$\Rightarrow$ ALG nu este $\frac{3}{2}$ aproximativ, e măcar $\frac{8}{5}$ aproximativ

VERTEX COVER

a) pt $(x_1 \vee x_2 \vee x_3) \wedge (x_4 \vee x_5 \vee x_3) \wedge \ldots \wedge (x_{n-1}, x_n, x_3)$

OPT = $\{\cancel{x_5}\}$, $\{x_3\}$

ALG = $n-1$

LUCAN CRISTIAN

VERTEX COVER

a) worst case: $(x_1 \vee x_2 \vee x_n) \wedge (x_2 \vee x_3 \vee x_n) \dots \wedge (x_{n-2} \vee x_{n-1} \vee x_n)$

$OPT = 1 (\{x_n\})$

$ALG = n-2 (\{x_1, x_2, \dots x_{n-2}\})$ (dacă se pot repeta variabilele în clauze mai apar și $\{x_{n-1} \vee x_n \vee x_n\} \wedge \{x_n \vee x_n \vee x_n\} \Rightarrow ALG = n$)

$\frac{ALG}{OPT} \approx n \Rightarrow ALG$ e ~~n(n-2)~~ n-aproximativ

b) ~~în loc să eliminăm o ce~~ când aleg o clauză, setez toate variabilele din ea pe true și elimin toate clauzele din care făceau parte variabilele.

fie C^* = mulț. de clauze ~~lor~~ variabile disjuncte

$(\forall) x \in Var(OPT)$, x corespunde cu cel mult o clauză din C^* $\Rightarrow$

$\Rightarrow OPT \geq |C^*| \, / \cdot 3$

$3OPT \geq 3|C^*| = ALG$ ~~(în ALG, la fiecare 3 variabile le corespunde maxim exact o clauză din C^*)~~

(fiecare triplet de variabile ales de ALG creează exact o clauză variabilă disjunctă)

$\Rightarrow ALG$ este 3-aproximativ

c) fie $f: X \to \mathbb{R}$, o fct. de cost pentru variabile

~~Pe lângă găsirea numărului minim de vars~~

~~Dorim să găsim costul minim minimizăm funcția~~ $\sum_{i=1}^{n} f(x_i)$

Dorim să găsim mulțimea de variabile care, dacă true, fac ca expresia să fie true și pt care $\sum_{x_i \in A} f(x_i)$ este minimă.

c) $A = \{a_1, a_2, \dots, a_m\}$

$a_i = 1$ dacă aleg variabila x_i
$a_i = 0$ altfel

Trebuie să minimizăm $\sum_{i=1}^{m} \cancel{f(x_i)} f(x_i) \cdot a_i$

Constrângeri: ~~$x_i + x$~~ pentru fiecare clauză $C = \{x_i \vee x_j \vee x_k\}$ avem
$a_i + a_j + a_k \geq 1$, $(\forall) i, j, k \in \{1, 2, \dots m\}$
$0 \leq a_i \leq 1$ $(\forall)$ $i \in \{1, 2, \dots m\}$

d) Pt că a_i trebuie să fie reale, vom ~~trimis~~ defini funcția

$g : A \to \{0, 1\}$, $g(a) = \begin{cases} 1, & a \geq \frac{1}{3} \\ 0, & a < \frac{1}{3} \end{cases}$ și vom încerca să aproximăm

$\underbrace{\sum_{i=1}^{m} f(x_i) \cdot a_i}_{OPT}$ ~~cu~~ $\sum_{i=1}^{m} f(x_i) \cdot g(a_i) = ALG$

$ALG \leq \sum_{i=1}^{m} f(x_i) \cdot 3a_i \leq 3 \sum_{i=1}^{m} f(x_i) \cdot a_i$

$\leq 3 OPT$ $\Rightarrow$ ALG este 3-aproximativ