Load Balance

1.

Răspuns a: Da, este posibil ca factorul de aproximare să fie 1,1

Exemplu: 60, 80, 60 (< 100)

Răspuns b: Nu, diferența dintre cele 2 mașini trebuie să fie de maxim 10 în cazul optim. O mașină poate avea încărcătura maximă de 105 și algoritmul nostru a dat 120

$\Rightarrow \frac{120}{105} = 1,14 > 1,1$

2. a. ALG1 - 2-aproximativ

ALG2 - 4-aproximativ

$ALG2(I) \geq 2 \cdot ALG1(I)$

$\forall_I$ pt $ALG1 \leq 2 * OPT$, pt $ALG2 \leq 4 * OPT$

Ppt. $ALG1 = 2 * OPT / \cdot 2$, $ALG2 < 4 * OPT$ R.

$\underbrace{ALG1 = \overset{4}{\overbrace{2 * 2}} * OPT \ , \ ALG2 < 4 * OPT}$

$\Rightarrow ALG2 < ALG1 \cdot 2 \Rightarrow$ Contradicție = Fals

b. $ALG1(I) > 2 \cdot ALG2(I)$

Pp. $ALG1 = 2 * OPT$ și $ALG2 = OPT$ (maxim și minim)

$\Rightarrow$ Adevărat pentru că $ALG1(I)$ poate fi $=$ cu $2 \cdot ALG2(I)$ dar nu poate fi $>$

3. Algoritmul Ordered - Scheduling Algorithm se poate îmbunătăți de la $\frac{3}{2}$ - aproximativ la $(\frac{3}{2} - \frac{1}{2m})$ - aprox.

k - indicele mașinii cu load maxim

q - ultima activitate adăugată la k

load'(k) - loadul mașinilor înainte ca q să fie asignat

$$ALG = load(k) = load'(k) + t_q \le \frac{1}{m} \sum_{1 \le j \le m} t_j + t_q$$

aici avem asignat și loadul q deci îl scădem

$$\le \frac{1}{m} \sum_{1 \le j \le m} (t_j - t_q) + t_q \le \frac{1}{m} \sum_{1 \le j \le m} t_j - \frac{1}{m} t_q + t_q \le$$

Algoritmul OSA $\Rightarrow t_q = \frac{1}{2}(t_m + t_{m+1}) = OPT$

$$\le \underbrace{\frac{1}{m} \sum_{1 \le j \le m} t_j}_{OPT} - \frac{1}{m} \cdot \underbrace{\frac{1}{2}(t_m + t_{m+1})}_{OPT} + \underbrace{\frac{1}{2}(t_m + t_{m+1})}_{OPT}$$

$$\le OPT - \frac{1}{2m} OPT + \frac{1}{2} OPT$$

$$\le (1 + \frac{1}{2} - \frac{1}{2m}) OPT \le (\frac{3}{2} - \frac{1}{2m}) OPT$$

TSP:

1. a. Toate muchiile au ponderea $\{1, 2\}$

Problema este NP-hard

R. A.

Pp că există un algoritm aproximativ a. î. pentru S- cost minim => algoritmul rezultă $c \cdot S$

Problema determinării HC în G ~~este~~ <u>nu</u> se poate rezolva în timp polinomial ①

Construim G' după G ($V(G) = V(G') = n$)

unde toate muchiile care apar în G' din G vin cu un cost de 1 sau 2

Completăm cu muchii până G' devine complet.

Muchii cu un cost $c * n$

Caz I:

Dacă G avea ciclu hamiltonian =>

=> Alg pt G' oferă un traseu de maxim $C * 2 * n$ (polinomial)

(în cazul în care toate muchiile sunt cu un cost 2)

Caz II:

Dacă G nu conținea ciclu hamiltonian =>

=> Alg pt G' conține cel mult $(n-1)$ muchii de cost 2

și o muchie de cost $c * n$ => Traseu optim $2(n-1) + c*n$.

=> Rezultatul algoritmului $> c * n$ (timp polinomial)

G' -> în timp polinomial
Alg -> în timp polinomial } => pot afla dacă G conține ciclu hamiltonian sau nu într-un timp polinomial (2)

Din (1) și (2) => Contradicție => Problema rămâne NP-hard și pentru costuri de 1 sau 2

b. Din lema 2: Inegalitatea triunghiului

$len((v_1, v_k)) \le len(v_1, \dots, v_k)$

În cazul de față, pentru ponderi de 1 sau 2, cel mai instabil caz este $K = 3$

$$len((\underbrace{v_1, v_3}_{2})) \le len(\underbrace{v_1, \underbrace{v_2}_{1}, v_3}_{1+1=2}) \leftarrow \text{minim posibil}$$

maxim posibil

=> Adevărat pt $K = 3$

Pentru $K > 3$ este clar că se respectă regula triunghiului.

TSP. 1. c.

Pp graful complet $G(V, E)$ cu toate muchiile de cost 1 => costul total pe TSP = n (nr. noduri)

MST din nodul 1 pe graful G conține $n-1$ muchii. Algoritmul parcurge MST de 2 ori de unde rezultă $2 * (n-1) = 2n - 2$

Pp $2n - 2 > \frac{3}{2} \cdot n$

$2n - \frac{3}{2} n > 2$

$4n - 3n > 4$

$n > 4$

=> $\forall n \in N > 4$ algoritmul nu este $\frac{3}{2}$ aproximativ