Load Balance

Problema 1

Fie o iterație a problemei Load Balancing (Cursul 2, slide-ul 16) pentru 2 mașini. La seminarul de algoritmi aproximativi unul dintre studenți propune un algoritm de rezolvare si sustine ca acesta este 1.1 aproximativ. El ruleaza algoritmul pe un set de n activitati si obtine o incarcatura de 80 pe una dintre masini, respectiv 120 pe cealalta. Este posibil ca factorul lui de aproximare sa fie corect?

a) tinand cont ca rezultatul obtinut anterior a fost făcut pe un set de activități, fiecare cu timpul de lucru cel mult 100 (0.5p)

b) tinand cont ca rezultatul obtinut anterior a fost făcut pe un set de activități, fiecare cu timpul de lucru cel mult 10 (0,5p)

Rezolvare:

a) Fie activitățile A_1 ($t_1 = 15$), A_2 ($t_2 = 25$), A_3 ($t_3 = 65$) și A_4 ($t_4 = 95$).

Programarea obținută de student este:

$$M1 \;-> \; A1 \,+\, A3 \,=\, 80$$

$$M2 \;-> \; A2 \,+\, A4 \,=\, 120$$

Programarea optima ar fi 110, după configurația:

$$M1 \;-> \; A1 \,+\, A4 \,=\, 110$$

$$M2 \;-> \; A2 \,+\, A3 \,=\, 90$$

Cum $ALG\ (I) \ = \ 120$ și $OPT\ (I) \ = \ 110$ => $OPT(I) \leq ALG(I) \leq ALG(I) * 1.1$.

Deci, este posibil ca factorul de aproximare să fie 1.1.

b) Având în considerare faptul că activitățile au o încarcătura mai mică sau egală cu 10, atunci diferența maximă la algoritmul optim va fi mai mică sau egală decât 10.

Cu o aproximare de 1.1, rezultă faptul că un algoritm 1.1 aproximativ ar avea diferența între încărcaturi cel mult egală cu 1.1*10=11.

Cum diferența dintre încarcături obținută de student este 120-80=40 > 11, ajungem la concluzia că algoritmul nu poate fi 1.1 aproximativ.

Problema 3

Fie algoritmul Ordered-Scheduling Algorithm (Cursul 2, slide-ul 42) care implica algoritmul descris anterior (slide 19) la care adăugăm o preprocesare cu care sortăm descrescător activitățile după timpul de desfășurare. Th. 2 afirmă că acest algoritm este 3/2 aproximativ. Arătați ca acest factor de aproximare poate fi îmbunătățit la 3/2-1/(2m). (2p)

Notații:

m – numărul de mașini

n – numărul de activități de procesat

t_j – unitatea de timp necesară pentru execuția activității

k – indicele mașinii cu load maxim în urma executării algoritmului

q – ultimul job adăugat mașinii k

load’(m) – load-ul mașinii m înainte de rapartizarea activității q

Ipoteză:

$$t_1 \geq t_2 \geq t_3 \geq \cdots \geq t_n$$

Leme

Lema 1.

$$OPT \geq max\left(\frac{1}{m}\sum_{1\leq j\leq n} t_j\,, max\,\{t_j \mid 1 \leq j \leq n\}\right).$$

Din ipoteză, Lema 1 devine:

$$OPT \geq max\left(\frac{1}{m}\sum_{1\leq j\leq n} t_j\,, t_1\right).$$

Lema 2. Dacă n>m, atunci

$$OPT \geq t_m + t_{m+1}$$

Rezolvare:

Din ipoteze rezultă că

$$ALG = load'(k) + t_q$$

Avem 2 cazuri:

1. $q \leq m$

   Atunci, fiecare activitate va fi repartizată altei mașini, deci

   $$ALG = t_q \leq t_1 \ (1)$$

   Din Lema 1 și (1), rezultă că $ALG \leq OPT$, deci $ALG = OPT$.

2. $q > m$

Atunci

$$load'(k) \leq \frac{1}{m} \sum_{1 \leq i \leq m} load'(i).$$

Continuând,

$$load'(k) \leq \frac{1}{m} \sum_{1 \leq i \leq m} load'(i) = \frac{1}{m} \sum_{1 \leq j < q} t_j \leq$$

$$\leq \frac{1}{m} \Big(\sum_{1 \leq j < q} t_j - t_q \Big) = \frac{1}{m} \sum_{1 \leq j < q} t_j - \frac{1}{m} t_q. \qquad (2)$$

Din Lema 1 și (2) rezultă că

$$load'(k) \leq OPT - \frac{1}{m} t_q =>$$

$$=> ALG = load'(k) + t_q \leq OPT - \frac{1}{m} t_q + t_q =>$$

$$=> ALG = load'(k) + t_q \leq OPT + \left(1 - \frac{1}{m}\right) t_q.$$

Din ipoteză avem $q > m$ și $t_1 \geq t_2 \geq t_3 \geq \cdots \geq t_n$, deci

$$t_q \leq \frac{t_m + t_{m+1}}{2} =>$$

$$=> ALG = load'(k) + t_q \leq OPT + \left(1 - \frac{1}{m}\right) \frac{t_m + t_{m+1}}{2}.$$

Aplicând Lema 2 pe relația anterioară obținem:

$$ALG \leq OPT + \left(1 - \frac{1}{m}\right)\frac{OPT}{2} =>$$

$$=> \ ALG \leq \left(\frac{3}{2} - \frac{1}{2m}\right) OPT =>$$

Algoritmul este $\frac{3}{2} - \frac{1}{2m}$ aproximativ.