Dumitru Florentin Giuliano
grupa 233

Load Balancing

1) a) Factorul dat de aproximare este bun.
Putem da un set de date de intrare pentru a obține această valoare (120 - 80).

Presupunem setul de activități: 80, 10, 10, 10, 10, 80.

Vom face algoritmul pe pași acum:

I) act: 80, 10, 10, 10, 10, 80
C1: 0
C2: 0

II) act: 10, 10, 10, 10, 80
C1: 80
C2: 0

III) act: 10, 10, 10, 80
C1: 80
C2: 10

IV) act: 10, 10, 80

C1: 80

C2: 20

V) act: 10, 80

C1: 80

C2: 30

VI) act: 80

C1: 80

C2: 50

VII) act: nu mai sunt

C1: 80

C2: 120

Deci am obținut 80 și 120.

~~2)~~ b) Factorul de aproximare nu este bun.
Diferența de 50 este prea mare deoarece valorile sunt cuprinse între ~~1~~ și 10.
~~Presupunem că~~

2) /ul ALG1 ~ 2 APROXIMATIV
ALG2 ~ 2APROXIMATIV

a) Este adevărat, deoarece ALG1 este 2-Aproximativ, adică că $OPT \leq ALG1 \leq 2OPT$, unde OPT este soluția optimă, adică cel mai bine aproximată

$ALG2 = 5$-Aproximativ $\Rightarrow OPT \leq ALG2 \leq 5 \cdot OPT$

Presupunem că avem inputul I astfel încât

$ALG1 = 2 * OPT$ și $ALG2 = 5 * OPT$

$\Rightarrow ALG2 = 2 * ALG1 \Rightarrow ALG2 \geq 2 ALG1$

b) Este fals, dar acest lucru îl vom demonstra prin reducere la absurd.

Presupunem că ar fi adevărat și că există un input I astfel încât $ALG1 \geq 2 * ALG2$

Știm că $OPT \leq ALG1 \leq 2OPT$ (1)

$OPT \leq ALG2 \leq 5 OPT$ (2)

Dacă $ALG1 \geq 2 ALG2$ (3)

$\Rightarrow$ din (1) U (2) U (3) $\Rightarrow$

$1/2 * OPT \leq ALG2 < OPT$ (F), deci nu există un astfel de I

3) Să zicem că r este indexul mașinii ce a fost adăugată ultima mașinii M. Vom demonstra că dacă $p_r \leq \frac{1}{2}$ OPT, atunci eroarea de mărginire nu poate fi depășită și dacă $p_r \geq \frac{1}{2}$ OPT, atunci împărțirea obținută este optimă.

Notațiile folosite în rezolvare:

J_i = jobul de index i

p_i = timpul de procesare a activității i

M_i = mașina de index i

a_i = timpul de terminare a activităților mașinii i

m = numărul de mașini

n = numărul de joburi

M = împărțirea obținută de acest algoritm

s_i = timpul de început al activității i

OPT = împărțirea folosind algoritmul optim

Algoritmul LPT are 2 pași:

a) Sortăm jobările descrescător

b) Punem succesiv jobările în prima mașină disponibilă.

Știm că $m \cdot OPT \geq (\sum a_i + \sum p_j)$

de aici obținem că $OPT \geq (\sum a_i + \sum p_j)/m$

Cum s_r se adaugă la mașina ce a terminat prima de executat o activitate =>

$$s_r \leq (\sum_{i=1}^{m} a_i + \sum_{i=j-1}^{r-1} p_i)/m \leq$$

$$\leq (m \cdot OPT - p_r)/m \leq OPT - \frac{p_r}{m}$$

I Presupunem că $p_r \leq 1/2\ OPT$, deci obținem că

$$M = s_r + p_r \leq OPT - \frac{p_r}{m} + p_r \leq$$

$$\leq OPT + (1-\frac{1}{m}) \cdot p_r \leq OPT + (1-\frac{1}{m}) \cdot \frac{OPT}{2}$$

$$= (\frac{3}{2} - \frac{1}{2m})OPT$$, deci marginea ~~super~~ dată nu poate fi depășită

II Presupunem că $p_r \geq \frac{OPT}{2}$

Acest lucru ar implica că în soluția optimă nicio mașină nu poate primi mai mult de o activitate ~~a~~ carei index este mai mic sau egal decât r, altfel soluția optimă SOL nu ar fi cea optimă.

Știm că $a_i \leq a_{i+1}$, $\forall i \leq n$
$\Rightarrow a_i \leq \frac{OPT}{2}$, $\forall i \leq r$

În acest timp avem activități cu timp de procesare mai mare decât OPT și $a_i \leq OPT$, pentru $\forall i \leq r$.

Algoritmul OPT ~~ii~~ va pune jobul J_i la mașina M_i cu $i \leq r$. Din definiția lui p_r, trebuie să fie adăugat singur la o mașină, deci împărțirea $M = a_r + p_r$, unde $a_r \leq OPT$.

Acum vom avea două cazuri posibile pentru a o demonstra (vom nota M^* împărțirea activităților pe fiecare caz):

a) există o mașină M_k, $k \geq r$, ce a primit jobul J_i cu $i \leq r$. În acest caz $M^* \geq a_k + p_i \geq a_r + p_r = M$

b) toate activitățile $\{J_1, \ldots, J_r\}$ au fost atribuite mașinilor $\{M_1, \ldots, M_{r-1}\}$.

În acest caz, există o mașină din $\{M_1, \dots, M_{k-1}\}$ astfel încât are cel puțin 2 activități din $\{J_1, \dots, J_k\}$, întrucât

$p_i \geq p_k \geq OPT > OPT$ $\Rightarrow$ $M' > a_k + p_k = M$.

7.