Tudorache Alexandru-Theodor

Grupa 242

Tema 1

Load Balance

1)

a)

Fie setul de date 80, 80, 40

Pentru acest exemplu, atat solutia optima, cat si cea propusa de student obtin incarcatura de 80 pe o masina si de 120 pe cealalta, astfel:

Masina 1: 80

Masina 2: 80, 40

Asadar, solutia studentului poate fi 1-aproximativa (adica optima), deci inclusiv 1.1 aproximativa.

b)

Cum una dintre masini are incarcatura de 120, pentru ca algoritmul sa fie 1.1 aproximativ, in solutia optima o masina ar trebui sa aiba incarcatura de minim $120/1.1=109.09\cong109$, asadar cealalta masina ar trebui sa aiba incarcatura de maxim $200-109=91$.

Presupunem prin absurd ca exista OPT o solutie optima cu incarcatura 109 pe masina 1 si 91 pe masina 2. Cum o activitate are timpul de lucru cel mult 10, daca extragem o activitate de pe masina 1 si o punem pe masina 2, in noua solutie pe masina 1 va fi o incarcatura cuprinsa intre 99 si 108, iar pe masina 2 o incarcatura cuprinsa intre 92 si 101, astfel obtinandu-se o solutie mai buna decat OPT ($108 < 109$; $101<109$). Absurd.

Daca presupunem ca exista o solutie optima cu o incarcatura mai mare decat 109 pe masina 1 si mai mica decat 91 pe masina 2, se va ajunge tot la o noua solutie optima, deoarece diferenta dintre incarcaturile masinilor va creste, astfel putandu-se obtine solutii din ce in ce mai bune decat cea din presupunere.

Deci, solutia optima nu poate contine o masina cu incarcatura de minim 109, astfel algoritmul propus de student nu poate fi 1.1 aproximativ.

3)

Fie k = indicele masinii cu load-ul maxim rezultat in urma executarii algoritmului

Deci solutia obtinuta de algoritm este ALG = load(k)

Fie j = ultima activitate adaugata pe masina k.

Fie load'(i) = load-ul masinii i fix inainte ca activitatea j sa fie pusa pe masina k, adica load-ul masinii i dupa ce au fost distribuite primele j-1 activitati.

Asadar, ALG = load(k) = load'(k) + t_j

Cum algoritmul selecteaza masina cu load minim si pune pe ea urmatoarea activitate, load'(k) ≤ load'(i), ∀ i ∈ {1, ..., m}

Deci load'(k) ≤ $\frac{1}{m}\sum_{1\le i\le m} load'(i) = \frac{1}{m}\sum_{1\le i\le j} t_i \le \frac{1}{m}\left(\sum_{1\le i\le n} t_i - t_j\right) \le \frac{1}{m}\sum_{1\le i\le n} t_i - \frac{t_j}{m} => load'(k) + t_j \le \frac{1}{m}\sum_{1\le i\le n} t_i + t_j(1-\frac{1}{m})$

- Daca j ≤ m, atunci activitatea j va fi pusa pe o masina goala, deci ALG = t_{max} ≤ OPT, unde t_{max} este activitatea cu timpul de lucru maxim. In acest caz, algoritmul este optim.
- Daca j > m, atunci:
$$load'(k) + t_j \le \frac{1}{m}\sum_{1\le i\le n} t_i + t_j\left(1-\frac{1}{m}\right) \le OPT + t_j(1-\frac{1}{m})$$
Cum activitatile sunt ordonate descrescator dupa timpul de lucru, iar j > m, atunci rezulta ca: $t_j \le \frac{1}{2}(t_m + t_{m+1})$, de unde $=> ALG \le OPT + \left(1-\frac{1}{m}\right) * \frac{1}{2}(t_m + t_{m+1})$
Din Lema 3(Curs 2, slide 43) avem $OPT \ge t_m + t_{m+1}$
$$=> ALG \le OPT + \left(\frac{1}{2} - \frac{1}{2m}\right) * OPT = \left(\frac{3}{2} - \frac{1}{2m}\right) * OPT$$
Deci, algoritmul dat este $(\frac{3}{2} - \frac{1}{2m})$ aproximativ.