Q0: In ce cazuri folosim algoritmi aproximativi?

In cazul in care ne confruntam cu o problema NP-hard si optam sa renuntam la gasirea solutiei optime, pentru a gasi totusi o solutie “acceptabila” dar in timp fezabil.

Q1: ce este factorul de aproximare pentru un algoritm?

Fie ALG - solutia noastra, OPT - solutia optima

In cazul unei probleme de minim, o constanta *c* se numeste factor de aproximare daca:

OPT≤ALG≤c\*OPT.

in cazul unei probleme de maxim

OPT≥ALG≥c\*OPT.

Q1.1 In cazul unei probleme de minim: Un algoritm 2-aproximativ poate fi numit 3-aproximativ?

Consecinta: pt un ALG sa gasim *c* - cat mai aproape de 1 (tight bound)

Q1.2 Cum putem sa justificam ca un *c* dat este tight bound?

In cazul problemelor de minim, ar tb sa gasim o intrare *I* astfel incat:

ALG(I) **=** c\*OPT(I)

Sau ar tb sa arat ca nu exista niciun c’<c cu c’ factor de aproximare pt ALG

1. Avem următorul scenariu: Avem *n* colete de transportat, fiecare avand greutatea de *w1, w2,...,wn.* Pentru a le transporta, putem folosi un număr de camioane, fiecare avand capacitatea de transport *G*. Presupunem că *wi≤G*, pentru orice *i*. Ne dorim sa minimizăm numărul de camioane folosite. Considerăm următorul plan de încărcare a camioanelor:

Odată ce avem la dispoziție un camion pt a fi încărcat, iterăm prin mulțimea coletelor, incărcându-le in camion, până când dăm peste primul colet ce nu mai incape. În acel moment considerat că am terminat de încărcat camionul curent și trecem la următorul camion, prima dată încărcând coletul care nu a mai încăput în cel precedent.

1. Arătați, printr-un exemplu simplu, că metoda de mai sus nu furnizează soluția optimă.
2. Arătați totuși că soluția de mai sus este un algoritm 2-aproximativ pentru problema noastră.

Raspuns:  
 a) G=100; W=(30,90,60)

ALG: (30);(90);(60)  
OPT: (90); (30;60)

b)   
Fie W - suma greutatilor tuturor coletelor;

OPT>=W/G

Cate camioane foloseste algoritmul nostru?

2 cazuri: daca algoritmul foloseste 2\*c camioane (numar par) sau 2\*c+1 (numar impar de camioane.

Luam cazul in care numarul de camioane folosite este impar.

Ce cantitate transporta cele 2c+1 camioane? W -cele 2c+1 camioane transporta intreaga cantitate de colete!

Fiecare pereche de camioane (dintre cele c perechi) ce cantitate minima transporta? G

fiecare pereche de camioane transporta o cantitate > G (evident din algoritm)

2\*c+1 > c\*G (cele 2\*c+1 camioane transporta o cantitate de colete > c\*G)

W>c\*G

OPT>=W/G>c  
OPT>c. Deci solutia optima este alcatuita din cel putin c+1 camioane.  
OPT>=c+1

2\*OPT>=2\*c+2>2c+1=ALG>=OPT

2) Dat fiind algoritmul Load-Balance (Cursul 2, slide 19) să se stabilească dacă următoarea afirmație este adevărată sau falsă.

”Pentru orice instanța a problemei de Load-Balace, exista o anumită ordine a procesării activităților astfel încât algoritmul de tip greedy să dea o soluție optimă”

Dacă afirmația este adevărată, oferiți o demonstrație, altfel, găsiți un contraexemplu.

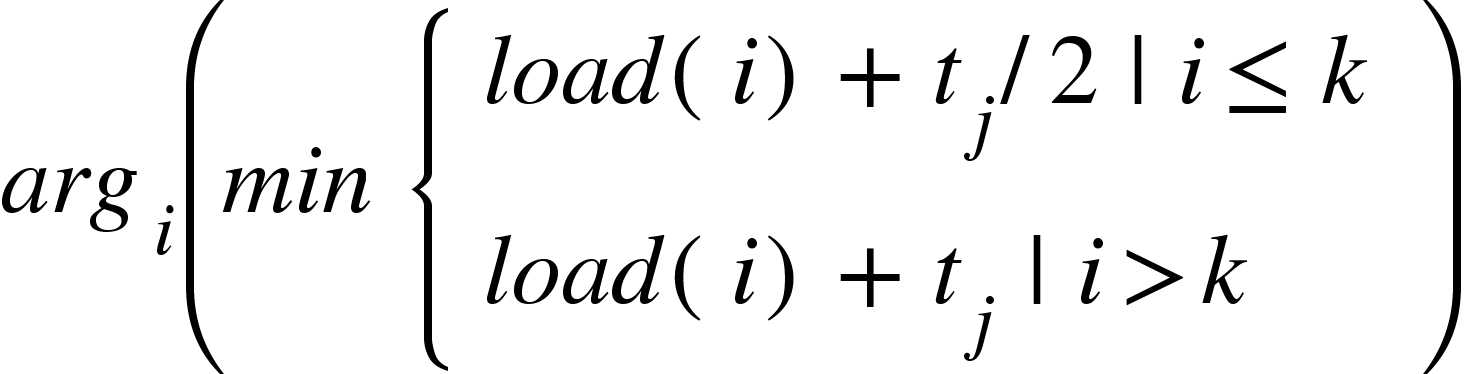
Da - mereu exista o ordonare astfel incat algoritmul descris in slide-ul 19 (care este in general un algoritm 2-aproximativ) va oferi mereu solutia optima.

3) Fie Problema Load Balance, dar cu următoarea modificare: Avem *n* joburi si *m* mașini, doar că pentru primele *k* mașini timpul de lucru al unei activități este înjumătățit. Să se găsească un algoritm bazat pe tehnica greedy care furnizeaza o soluție de cel mult 3xOPT.

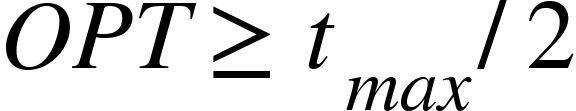
Raspuns:

iteram prin multimea de joburi

jobul j il punem pe masina cu proprietatea ca termina j cel mai repede:



Demonstratie ca algoritmul schitat mai sus este 3 aproximativ:

Trebuie sa stabilim niste lower bounds pt optim:  


<math xmlns="http://www.w3.org/1998/Math/MathML"><mi>O</mi><mi>P</mi><mi>T</mi><mo>&#x2265;</mo><mfrac><mn>1</mn><mrow><mi>m</mi><mo>-</mo><mi>k</mi><mo>+</mo><mn>2</mn><mi>k</mi></mrow></mfrac><munder><mo>&#x2211;</mo><mrow><mn>1</mn><mo>&#x2264;</mo><mi>i</mi><mo>&#x2264;</mo><mi>n</mi></mrow></munder><msub><mi>t</mi><mi>i</mi></msub><mspace linebreak="newline"/><mi>J</mi><mi>u</mi><mi>s</mi><mi>t</mi><mi>i</mi><mi>f</mi><mi>i</mi><mi>c</mi><mi>a</mi><mi>r</mi><mi>e</mi><mo>:</mo><mspace linebreak="newline"/><mi>N</mi><mi>o</mi><mi>t</mi><mi>a</mi><mi>m</mi><mo>&#xA0;</mo><mi>t</mi><mo>=</mo><mfrac><mn>1</mn><mrow><mi>m</mi><mo>-</mo><mi>k</mi><mo>+</mo><mn>2</mn><mi>k</mi></mrow></mfrac><munder><mo>&#x2211;</mo><mrow><mn>1</mn><mo>&#x2264;</mo><mi>i</mi><mo>&#x2264;</mo><mi>n</mi></mrow></munder><msub><mi>t</mi><mi>i</mi></msub><mspace linebreak="newline"/><mi>A</mi><mi>f</mi><mi>i</mi><mi>r</mi><mi>m</mi><mi>a</mi><mi>t</mi><mi>i</mi><mi>e</mi><mo>:</mo><mo>&#xA0;</mo><mi>n</mi><mi>u</mi><mo>&#xA0;</mo><mi>s</mi><mi>e</mi><mo>&#xA0;</mo><mi>p</mi><mi>o</mi><mi>a</mi><mi>t</mi><mi>e</mi><mo>&#xA0;</mo><mi>c</mi><mi>a</mi><mo>&#xA0;</mo><mi>t</mi><mi>o</mi><mi>t</mi><mi>e</mi><mo>&#xA0;</mo><mi>m</mi><mi>a</mi><mi>sin</mi><mi>i</mi><mi>l</mi><mi>e</mi><mo>&#xA0;</mo><mi>s</mi><mi>a</mi><mo>&#xA0;</mo><mi>l</mi><mi>u</mi><mi>c</mi><mi>r</mi><mi>e</mi><mi>z</mi><mi>e</mi><mo>&#xA0;</mo><mi>m</mi><mi>a</mi><mi>i</mi><mo>&#xA0;</mo><mi>p</mi><mi>u</mi><mi>t</mi><mi>i</mi><mi>n</mi><mo>&#xA0;</mo><mi>d</mi><mi>e</mi><mi>c</mi><mi>a</mi><mi>t</mi><mo>&#xA0;</mo><mi>t</mi><mo>&#xA0;</mo><mi>u</mi><mi>n</mi><mi>i</mi><mi>t</mi><mi>a</mi><mi>t</mi><mi>i</mi><mo>&#xA0;</mo><mi>d</mi><mi>e</mi><mo>&#xA0;</mo><mi>t</mi><mi>i</mi><mi>m</mi><mi>p</mi><mo>&#xA0;</mo><mi>s</mi><mi>i</mi><mo>&#xA0;</mo><mi>s</mi><mi>a</mi><mo>&#xA0;</mo><mi>i</mi><mi>s</mi><mi>i</mi><mo>&#xA0;</mo><mi>t</mi><mi>e</mi><mi>r</mi><mi>m</mi><mi>i</mi><mi>n</mi><mi>e</mi><mo>&#xA0;</mo><mi>t</mi><mi>o</mi><mi>a</mi><mi>t</mi><mi>e</mi><mo>&#xA0;</mo><mi>a</mi><mi>c</mi><mi>t</mi><mi>i</mi><mi>v</mi><mi>i</mi><mi>t</mi><mi>a</mi><mi>t</mi><mi>i</mi><mi>l</mi><mi>e</mi><mspace linebreak="newline"/><mi>P</mi><mi>P</mi><mo>&#xA0;</mo><mi>c</mi><mi>a</mi><mo>&#xA0;</mo><mi>t</mi><mi>o</mi><mi>a</mi><mi>t</mi><mi>e</mi><mo>&#xA0;</mo><mi>m</mi><mi>a</mi><mi>sin</mi><mi>i</mi><mi>l</mi><mi>e</mi><mo>&#xA0;</mo><mi>l</mi><mi>u</mi><mi>c</mi><mi>r</mi><mi>e</mi><mi>a</mi><mi>z</mi><mi>a</mi><mo>&#xA0;</mo><mi>m</mi><mi>a</mi><mi>i</mi><mo>&#xA0;</mo><mi>p</mi><mi>u</mi><mi>t</mi><mi>i</mi><mi>n</mi><mo>&#xA0;</mo><mi>d</mi><mi>e</mi><mo>&#xA0;</mo><mi>t</mi><mo>&#xA0;</mo><mi>u</mi><mi>n</mi><mi>i</mi><mi>t</mi><mi>a</mi><mi>t</mi><mi>i</mi><mo>&#xA0;</mo><mi>s</mi><mi>i</mi><mo>&#xA0;</mo><mi>t</mi><mi>o</mi><mi>t</mi><mi>u</mi><mi>s</mi><mi>i</mi><mo>&#xA0;</mo><mi>f</mi><mi>i</mi><mi>n</mi><mi>a</mi><mi>l</mi><mi>i</mi><mi>z</mi><mi>e</mi><mi>a</mi><mi>z</mi><mi>a</mi><mo>&#xA0;</mo><mi>t</mi><mi>o</mi><mi>a</mi><mi>t</mi><mi>e</mi><mo>&#xA0;</mo><mi>a</mi><mi>c</mi><mi>t</mi><mi>i</mi><mi>v</mi><mi>i</mi><mi>t</mi><mi>a</mi><mi>t</mi><mi>i</mi><mi>l</mi><mi>e</mi><mspace linebreak="newline"/><munder><mo>&#x2211;</mo><mrow><mn>1</mn><mo>&#x2264;</mo><mi>i</mi><mo>&#x2264;</mo><mi>n</mi></mrow></munder><msub><mi>t</mi><mi>i</mi></msub><mo>=</mo><mfenced><mrow><mi>m</mi><mo>-</mo><mi>k</mi><mo>+</mo><mn>2</mn><mi>k</mi></mrow></mfenced><mo>&#xB7;</mo><mi>t</mi><mo>=</mo><mfenced><mrow><mi>m</mi><mo>-</mo><mi>k</mi></mrow></mfenced><mi>t</mi><mo>+</mo><mi>k</mi><mo>&#xB7;</mo><mn>2</mn><mi>t</mi><mo>&gt;</mo><munder><mo>&#x2211;</mo><mrow><mn>1</mn><mo>&#x2264;</mo><mi>i</mi><mo>&#x2264;</mo><mi>n</mi></mrow></munder><msub><mi>t</mi><mi>i</mi></msub><mo>&#xA0;</mo><mo>&#xA0;</mo><mi>c</mi><mi>o</mi><mi>n</mi><mi>t</mi><mi>r</mi><mi>a</mi><mi>d</mi><mi>i</mi><mi>c</mi><mi>t</mi><mi>i</mi><mi>e</mi><mo>!</mo><mspace linebreak="newline"/><mi>i</mi><mi>n</mi><mi>s</mi><mi>e</mi><mi>a</mi><mi>m</mi><mi>n</mi><mi>a</mi><mo>&#xA0;</mo><mi>c</mi><mi>a</mi><mo>&#xA0;</mo><mi>n</mi><mi>u</mi><mo>&#xA0;</mo><mi>s</mi><mi>e</mi><mo>&#xA0;</mo><mi>p</mi><mi>o</mi><mi>a</mi><mi>t</mi><mi>e</mi><mo>&#xA0;</mo><mi>c</mi><mi>a</mi><mo>&#xA0;</mo><mi>t</mi><mi>o</mi><mi>a</mi><mi>t</mi><mi>e</mi><mo>&#xA0;</mo><mi>m</mi><mi>a</mi><mi>sin</mi><mi>i</mi><mi>l</mi><mi>e</mi><mo>&#xA0;</mo><mi>s</mi><mi>a</mi><mo>&#xA0;</mo><mi>l</mi><mi>u</mi><mi>c</mi><mi>r</mi><mi>e</mi><mi>z</mi><mi>e</mi><mo>&#xA0;</mo><mi>m</mi><mi>a</mi><mi>i</mi><mo>&#xA0;</mo><mi>p</mi><mi>u</mi><mi>t</mi><mi>i</mi><mi>n</mi><mo>&#xA0;</mo><mi>d</mi><mi>e</mi><mo>&#xA0;</mo><mi>t</mi><mo>&#xA0;</mo><mi>u</mi><mi>n</mi><mi>i</mi><mi>t</mi><mi>a</mi><mi>t</mi><mi>i</mi><mspace linebreak="newline"/><mi>O</mi><mi>P</mi><mi>T</mi><mo>&#x2265;</mo><mi>t</mi></math>

<math xmlns="http://www.w3.org/1998/Math/MathML"><mi>f</mi><mi>i</mi><mi>e</mi><mo>&#xA0;</mo><mi>q</mi><mo>&#xA0;</mo><mi>a</mi><mi>c</mi><mi>t</mi><mi>i</mi><mi>v</mi><mi>i</mi><mi>t</mi><mi>a</mi><mi>t</mi><mi>e</mi><mi>a</mi><mo>&#xA0;</mo><mi>c</mi><mi>a</mi><mi>r</mi><mi>e</mi><mo>&#xA0;</mo><mi>v</mi><mi>a</mi><mo>&#xA0;</mo><mi>d</mi><mi>a</mi><mo>&#xA0;</mo><mi>t</mi><mi>i</mi><mi>m</mi><mi>p</mi><mi>u</mi><mi>l</mi><mo>&#xA0;</mo><mi>m</mi><mi>a</mi><mi>x</mi><mi>i</mi><mi>m</mi><mo>&#xA0;</mo><mi>d</mi><mi>e</mi><mo>&#xA0;</mo><mi>d</mi><mi>e</mi><mi>s</mi><mi>f</mi><mi>a</mi><mi>s</mi><mi>u</mi><mi>r</mi><mi>a</mi><mi>r</mi><mi>e</mi><mo>&#xA0;</mo><mi>p</mi><mi>e</mi><mo>&#xA0;</mo><mi>o</mi><mo>&#xA0;</mo><mi>m</mi><mi>a</mi><mi>sin</mi><mi>a</mi><mspace linebreak="newline"/><mi>f</mi><mi>i</mi><mi>e</mi><mo>&#xA0;</mo><mi>p</mi><mo>&#xA0;</mo><mi>m</mi><mi>a</mi><mi>sin</mi><mi>a</mi><mo>&#xA0;</mo><mi>p</mi><mi>e</mi><mi>s</mi><mi>t</mi><mi>e</mi><mo>&#xA0;</mo><mi>c</mi><mi>a</mi><mi>r</mi><mi>e</mi><mo>&#xA0;</mo><mi>v</mi><mi>a</mi><mo>&#xA0;</mo><mi>s</mi><mi>o</mi><mi>s</mi><mi>i</mi><mo>&#xA0;</mo><mi>a</mi><mi>c</mi><mi>t</mi><mi>i</mi><mi>v</mi><mi>i</mi><mi>t</mi><mi>a</mi><mi>t</mi><mi>e</mi><mi>a</mi><mo>&#xA0;</mo><mi>q</mi><mspace linebreak="newline"/><mi>l</mi><mi>o</mi><mi>a</mi><mi>d</mi><mo>'</mo><mfenced><mi>p</mi></mfenced><mo>&#xA0;</mo><mo>-</mo><mo>&#xA0;</mo><mi>l</mi><mi>o</mi><mi>a</mi><mi>d</mi><mi>u</mi><mi>l</mi><mo>&#xA0;</mo><mi>l</mi><mi>u</mi><mi>i</mi><mo>&#xA0;</mo><mi>p</mi><mo>&#xA0;</mo><mi>i</mi><mi>n</mi><mi>a</mi><mi>i</mi><mi>n</mi><mi>t</mi><mi>e</mi><mo>&#xA0;</mo><mi>s</mi><mi>a</mi><mo>&#xA0;</mo><mi>s</mi><mi>o</mi><mi>s</mi><mi>e</mi><mi>a</mi><mi>s</mi><mi>c</mi><mi>a</mi><mo>&#xA0;</mo><mi>m</mi><mi>a</mi><mi>sin</mi><mi>a</mi><mo>&#xA0;</mo><mi>q</mi><mspace linebreak="newline"/><mi>A</mi><mi>L</mi><mi>G</mi><mo>=</mo><mo>&#xA0;</mo><mi>l</mi><mi>o</mi><mi>a</mi><mi>d</mi><mo>'</mo><mfenced><mi>p</mi></mfenced><mo>+</mo><mfenced open="{" close=""><mtable columnalign="left"><mtr><mtd><msub><mi>t</mi><mi>q</mi></msub><mo>&#xA0;</mo><mo>|</mo><mo>&#xA0;</mo><mi>p</mi><mo>&gt;</mo><mi>k</mi></mtd></mtr><mtr><mtd><msub><mi>t</mi><mi>q</mi></msub><mo>/</mo><mn>2</mn><mo>&#xA0;</mo><mo>|</mo><mo>&#xA0;</mo><mi>p</mi><mo>&#x2264;</mo><mi>k</mi></mtd></mtr></mtable></mfenced><mspace linebreak="newline"/><mi>A</mi><mi>L</mi><mi>G</mi><mo>&#x2264;</mo><mo>&#xA0;</mo><mi>l</mi><mi>o</mi><mi>a</mi><mi>d</mi><mo>'</mo><mfenced><mi>p</mi></mfenced><mo>+</mo><msub><mi>t</mi><mi>q</mi></msub><mo>&#x2264;</mo><mfrac><mn>1</mn><mrow><mi>m</mi><mo>-</mo><mi>k</mi><mo>+</mo><mn>2</mn><mi>k</mi></mrow></mfrac><munder><mrow><mo>&#x2211;</mo><msub><mi>t</mi><mi>j</mi></msub></mrow><mrow><mn>1</mn><mo>&#x2264;</mo><mi>j</mi><mo>&#x2264;</mo><mi>n</mi></mrow></munder><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>+</mo><mo>&#xA0;</mo><mn>2</mn><mo>&#xB7;</mo><mfrac><msub><mi>t</mi><mi>q</mi></msub><mn>2</mn></mfrac><mo>&#x2264;</mo><mfrac><mn>1</mn><mrow><mi>m</mi><mo>-</mo><mi>k</mi><mo>+</mo><mn>2</mn><mi>k</mi></mrow></mfrac><munder><mrow><mo>&#x2211;</mo><msub><mi>t</mi><mi>j</mi></msub></mrow><mrow><mn>1</mn><mo>&#x2264;</mo><mi>j</mi><mo>&#x2264;</mo><mi>n</mi></mrow></munder><mo>&#xA0;</mo><mo>&#xA0;</mo><mo>+</mo><mo>&#xA0;</mo><mn>2</mn><mo>&#xB7;</mo><mfrac><msub><mi>t</mi><mrow><mi>m</mi><mi>a</mi><mi>x</mi></mrow></msub><mn>2</mn></mfrac><mspace linebreak="newline"/><mi>A</mi><mi>L</mi><mi>G</mi><mo>&#x2264;</mo><mi>L</mi><mi>B</mi><mo>+</mo><mn>2</mn><mo>&#xB7;</mo><mi>L</mi><mi>B</mi><mo>=</mo><mn>3</mn><mo>&#xB7;</mo><mi>L</mi><mi>B</mi><mo>&#x2264;</mo><mn>3</mn><mo>&#xB7;</mo><mi>O</mi><mi>P</mi><mi>T</mi></math>