**Problema rucsacului - varianta discretă (nefracționară)- 1/0 Knapsack Problem**

**Ipoteza de lucru:**

Avem *n* obiecte oi, fiecare cu o greutate și o valoare. Și un rucsac de capacitate G. Pentru orice obiect oi avem iar

**Pseudocod:**

Fie **L** - lista obiectelor ***sortate descrescător*** după raportul **valoare/greutate**

Fie **Op** - **obiectul cu profitul maxim** din lista de obiecte

S=0, G = capacitatea rucsacului.

Pentru fiecare obiect O din L

Dacă

**ALG = max (*val*(S), Op)**

**Corectitudine:**

În primul rând, este evident că algoritmul de mai sus ne oferă o soluție fezabilă. Elementele care au ca suma valorilor S vor avea o greutate totală ≤ capacitatea rucsacului, respectiv Op încape și el în rucsac de unul singur.

**Trebuie să justificăm doar factorul de aproximare.**

Fie OPT1/0 valoarea optima pentru Problema Rucsacului in varianta 1/0, respectiv OPTG valoarea optimă, furnizată de algoritmul de tip greedy pentru problema rucsacului in varianta în care aveam voie să ”tăiem” obiecte pentru a le încărca în rucsac.

**Cum este OPT1/0 față de OPTG ?**

**Avem:**

<math xmlns="http://www.w3.org/1998/Math/MathML"><mi>O</mi><mi>P</mi><msub><mi>T</mi><mrow><mn>1</mn><mo>/</mo><mn>0</mn></mrow></msub><mo>&#x2264;</mo><mo>&#xA0;</mo><mi>O</mi><mi>P</mi><msub><mi>T</mi><mi>G</mi></msub><mspace linebreak="newline"/><mi>F</mi><mi>i</mi><mi>e</mi><mo>&#xA0;</mo><mi>k</mi><mo>&#xA0;</mo><mi>i</mi><mi>n</mi><mi>d</mi><mi>i</mi><mi>c</mi><mi>e</mi><mi>l</mi><mi>e</mi><mo>&#xA0;</mo><mi>p</mi><mi>r</mi><mi>i</mi><mi>m</mi><mi>u</mi><mi>l</mi><mi>u</mi><mi>i</mi><mo>&#xA0;</mo><mi>o</mi><mi>b</mi><mi>i</mi><mi>e</mi><mi>c</mi><mi>t</mi><mo>&#xA0;</mo><mi>c</mi><mi>a</mi><mi>r</mi><mi>e</mi><mo>&#xA0;</mo><mi>n</mi><mi>u</mi><mo>&#xA0;</mo><mi>e</mi><mi>s</mi><mi>t</mi><mi>e</mi><mo>&#xA0;</mo><mi>a</mi><mi>d</mi><mi>a</mi><mi>u</mi><mi>g</mi><mi>a</mi><mi>t</mi><mo>&#xA0;</mo><mi>i</mi><mi>n</mi><mo>&#xA0;</mo><mi>a</mi><mi>l</mi><mi>g</mi><mi>o</mi><mi>r</mi><mi>i</mi><mi>t</mi><mi>m</mi><mi>u</mi><mi>l</mi><mo>&#xA0;</mo><mi>d</mi><mi>e</mi><mo>&#xA0;</mo><mi>l</mi><mi>a</mi><mo>&#xA0;</mo><mi>i</mi><mi>n</mi><mi>c</mi><mi>e</mi><mi>p</mi><mi>u</mi><mi>t</mi><mi>u</mi><mi>l</mi><mo>&#xA0;</mo><mi>p</mi><mi>a</mi><mi>g</mi><mi>i</mi><mi>n</mi><mi>i</mi><mi>i</mi><mo>.</mo><mspace linebreak="newline"/><mi>O</mi><mi>P</mi><msub><mi>T</mi><mrow><mn>1</mn><mo>/</mo><mn>0</mn></mrow></msub><mo>&#x2264;</mo><mi>O</mi><mi>P</mi><msub><mi>T</mi><mi>G</mi></msub><mo>&#xA0;</mo><mo>&#x2264;</mo><mo>&#xA0;</mo><munder><mo>&#x2211;</mo><mrow><mn>1</mn><mo>&#x2264;</mo><mi>i</mi><mo>&#x2264;</mo><mi>k</mi></mrow></munder><mi>v</mi><mi>a</mi><mi>l</mi><mfenced><msub><mi>O</mi><mi>i</mi></msub></mfenced><mo>=</mo><munder><mo>&#x2211;</mo><mrow><mn>1</mn><mo>&#x2264;</mo><mi>i</mi><mo>&lt;</mo><mi>k</mi></mrow></munder><mi>v</mi><mi>a</mi><mi>l</mi><mfenced><msub><mi>O</mi><mi>i</mi></msub></mfenced><mo>+</mo><mi>v</mi><mi>a</mi><mi>l</mi><mfenced><msub><mi>O</mi><mi>k</mi></msub></mfenced><mo>&#x2264;</mo><munder><mo>&#x2211;</mo><mrow><mn>1</mn><mo>&#x2264;</mo><mi>i</mi><mo>&lt;</mo><mi>k</mi></mrow></munder><mi>v</mi><mi>a</mi><mi>l</mi><mfenced><msub><mi>O</mi><mi>i</mi></msub></mfenced><mo>+</mo><mi>v</mi><mi>a</mi><mi>l</mi><mfenced><msub><mi>O</mi><mi>p</mi></msub></mfenced><mspace linebreak="newline"/><mi>A</mi><mi>L</mi><mi>G</mi><mo>=</mo><mi>m</mi><mi>a</mi><mi>x</mi><mfenced><mrow><mi>S</mi><mo>,</mo><msub><mi>O</mi><mi>p</mi></msub></mrow></mfenced><mspace linebreak="newline"/><mi>O</mi><mi>P</mi><msub><mi>T</mi><mrow><mn>1</mn><mo>/</mo><mn>0</mn></mrow></msub><mo>&#x2264;</mo><munder><mo>&#x2211;</mo><mrow><mn>1</mn><mo>&#x2264;</mo><mi>i</mi><mo>&lt;</mo><mi>k</mi></mrow></munder><mi>v</mi><mi>a</mi><mi>l</mi><mfenced><msub><mi>O</mi><mi>i</mi></msub></mfenced><mo>+</mo><mi>v</mi><mi>a</mi><mi>l</mi><mfenced><msub><mi>O</mi><mi>p</mi></msub></mfenced><mo>&#x2264;</mo><mi>A</mi><mi>L</mi><mi>G</mi><mo>&#xA0;</mo><mo>+</mo><mo>&#xA0;</mo><mi>A</mi><mi>L</mi><mi>G</mi><mo>=</mo><mn>2</mn><mo>&#xB7;</mo><mi>A</mi><mi>L</mi><mi>G</mi><mspace linebreak="newline"/><mi>O</mi><mi>P</mi><msub><mi>T</mi><mrow><mn>1</mn><mo>/</mo><mn>0</mn></mrow></msub><mo>&#x2264;</mo><mn>2</mn><mo>&#xB7;</mo><mi>A</mi><mi>L</mi><mi>G</mi></math>  
  
Ex intrare pt care abaterea e maxima

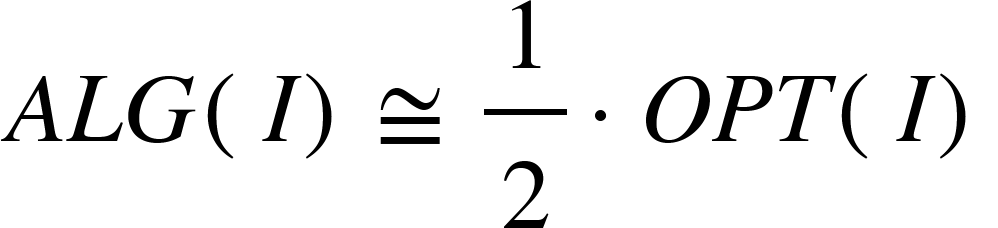
G=100

Ob (val/greutate)=[(50+eps1)/(50+eps2), 50/50, 50/50]

cu eps1>eps2>0

Evident profitul maxim este 100

profitul solutiei algoritmului este 50+eps1



deci ½ este un “tight upper bound”