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

Ipoteza de lucru:

Avem *n* obiecte o_i, fiecare cu o greutate și o valoare. Și un rucsac de capacitate G. Pentru orice obiect o_i avem $greutate(o_i) \leq G$ iar $\sum_{o \in O} greutate(o) > G$

Pseudocod:

Fie **L** - lista obiectelor ***sortate descrescător*** după raportul **valoare/greutate**
Fie $\mathbf{O_p}$ - **obiectul cu profitul maxim** din lista de obiecte
S=0, G = capacitatea rucsacului.
Pentru fiecare obiect O din L
    Dacă $greutate(O) \leq G,\ atunci\ S \mathrel{+}= val(O),\ G \mathrel{-}= greutate(O)$
ALG = max (*val*(S), $\mathbf{O_p}$)

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 O_p încape și el în rucsac de unul singur.

Trebuie să justificăm doar factorul de aproximare.

Fie $OPT_{1/0}$ valoarea optima pentru Problema Rucsacului in varianta 1/0, respectiv OPT_G 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 $OPT_{1/0}$ față de OPT_G ?

$OPT_{1/0} \leq OPT_G$

Avem:

$$OPT_{1/0} \leq OPT_G$$

Fie k indicele primului obiect care nu este adaugat in algoritmul de la inceputul paginii.

$$OPT_{1/0} \leq OPT_G \leq \sum_{1 \leq i \leq k} val(O_i) = \sum_{1 \leq i < k} val(O_i) + val(O_k) \leq \sum_{1 \leq i < k} val(O_i) + val(O_p)$$

$$ALG = max(S, O_p)$$

$$OPT_{1/0} \leq \sum_{1 \leq i < k} val(O_i) + val(O_p) \leq ALG + ALG = 2 \cdot ALG$$

$$OPT_{1/0} \leq 2 \cdot ALG$$

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

$$ALG(I) \cong \frac{1}{2} \cdot OPT(I)$$

deci ½ este un “tight upper bound”