Fundamental Computer Science
Studying the KnapSack problem

Denis Trystram
MoSIG1 and M1Info – University Grenoble-Alpes

April, 2021

$\{0, 1\}$-KnapSack

Decision form of the problem:

Knapsack

Input: a set of items A, two positive integers $K, W \in \mathbb{N}$, and for each $a \in A$ a profit $p(a) \in \mathbb{N}$ and a weight $w(a) \in \mathbb{N}$

Question: is there a set $B \subseteq A$ such that

$$\sum_{a \in B} p(a) \geq K \text{ and } \sum_{a \in B} w(a) \leq W \text{ ?}$$

We will use $p(a)$ or simply p_i for item i

Show that this problem is in NP-COMPLETE.

Tip: Easy by using a reduction from 2-PARTITION or SUBSETSUM.

Easy.

Dynamic Programming

Write the Dynamic Programming algorithm for solving this problem.

- $P(1, w) = 0$ if the item does not fit $w < \omega_1$ otherwise $P(1, w) = p_1$
- $P(k, w) = P(k-1, w)$ if $w < \omega_k$ and $max(P(k-1, w), P(k-1, w-1) + p_k)$

where $P(k, w)$ is the best possible profit to select the first k items in a sack of capacity w.

We target $P(K, W)$

Example

Let consider the optimization version of the problem.

A capacity $W = 8$ with 7 items (weight,profit):
$(5,2) \quad (8,3) \quad (11,6) \quad (9,4) \quad (12,5) \quad (17,8) \quad (8,4)$

Let us introduce the *density* of item a as the ratio $\lambda(a) = \frac{p(a)}{w(a)}$

A first attempt for solving the problem

Greedy algorithm, highest density first.

- Sort the items by non-increasing density:

$(8,3)$	$(5,2)$	$(12,5)$	$(9,4)$	$(17,8)$	$(8,4)$	$(11,6)$
2.67	2.5	2.4	2.25	2.12	2	1.67

- Select one item after the other in this order.
- What is the cost of the previous instance?
- Is it optimal?
- Is the algorithm guarantied by an approximation?

A first attempt for solving the problem

Greedy algorithm, highest density first.

- Sort the items by non-increasing density:

$(8,3)$	$(5,2)$	$(12,5)$	$(9,4)$	$(17,8)$	$(8,4)$	$(11,6)$
2.67	2.5	2.4	2.25	2.12	2	1.67

- Select one item after the other in this order.
- What is the cost of the previous instance?
- Is it optimal?
- Is the algorithm guarantied by an approximation?
- **NO!** Show that this process is arbitrarily bad.

Bad instances

Let denote by $f_{BD}(\mathcal{I})$ the profit of a solution obtained by the naive strategy *BestDensity* on the instance $\mathcal{I}$, and $OPT(\mathcal{I})$ the optimal profit for this instance.

$n = 2$ items and $W = k$.

- $(p_1 = 1, w_1 = 1)$
- $(p_2 = k - 1, w_2 = k)$

Proof

- The density of the first item is 1
- The density of the second one is $\frac{k-1}{k} = 1 - \frac{1}{k}$.

The density of the first item is larger, thus, we select this one first.
The second item can no more fit into the sack.
Thus, the profit is equal to 1.
If we have selected the second first, the profit would be $k - 1$ (in $\Theta(k)$)

This strategy is arbitrarily bad as k can be taken as large as we wish...

Improvement

We modifiy the previous heuristic in order to obtain a guarantee

- Sort the items by non-increasing density.
- Build a solution greedily as before until the next item does not fit into the sack (call it $\mathcal{S}_1$).
- Consider the alternative solution $\mathcal{S}_2$ composed of the item with the largest profit.
- Take the maximum between $\mathcal{S}_1$ and $\mathcal{S}_2$.

Show that

This policy is a $\frac{1}{2}$-approximation.

Proof

We show that this algorithm, called A, is never worse than twice the optimal.

- For each instance $\mathcal{I}$:
  $2 \cdot f_A(\mathcal{I}) \geq OPT(\mathcal{I})$
  where $f_A(\mathcal{I})$ is the profit of the solution obtained by the algorithm.

Proof (cont'd)

We have the following property:

$OPT \leq \Sigma_{i \in S_1} p_i + p_{i_T}$
that corresponds to: $OPT \leq \Sigma_{i \in S_1} p_i + p'_{i_T}$ where p'_{i_T} is the part of the truncated item i_T that saturates the sack.
Of course, it is lower than p_{i_T}.

This policy is a $\frac{1}{2}$-approximation.

The cost is $f_A = max(p_{i_T}, \Sigma_{i \in S_1} p_i)$
as $2.max(x, y) \geq x + y$, we have:
$2f_A \geq \Sigma_{i \in S_1} p_i + p_{i_T} \geq OPT$.