Greedy Algorithms

Fractional Knapsack

The Knapsack Problem

A	B	C
$100	$10	$120
2 pd	2pd	3 pd

Capacity of knapsack: $K = 4$

Fractional Knapsack Problem:
Can take a fraction of an item.

Solution:

2 pd A $100	2 pd C $80

0-1 Knapsack Problem:
Can only take or leave item. You can't take a fraction.

Solution:

3 pd C $120

Formal Definition

Given a knapsack of capacity K and n items of weights W and value/profit V, we have to give a solution of the form:

X = {x1, x2, x3, ,,,, , xn}   0 <= xi <= 1

weight	w_1	w_2	$\dots$	w_n
value	v_1	v_2	$\dots$	v_n

Find: $0 \leq x_i \leq 1$, $i = 1, 2, \dots, n$ such that

$$\sum_{i=1}^{n} x_i w_i \leq K$$

and the following is maximized:

$$\sum_{i=1}^{n} x_i v_i$$

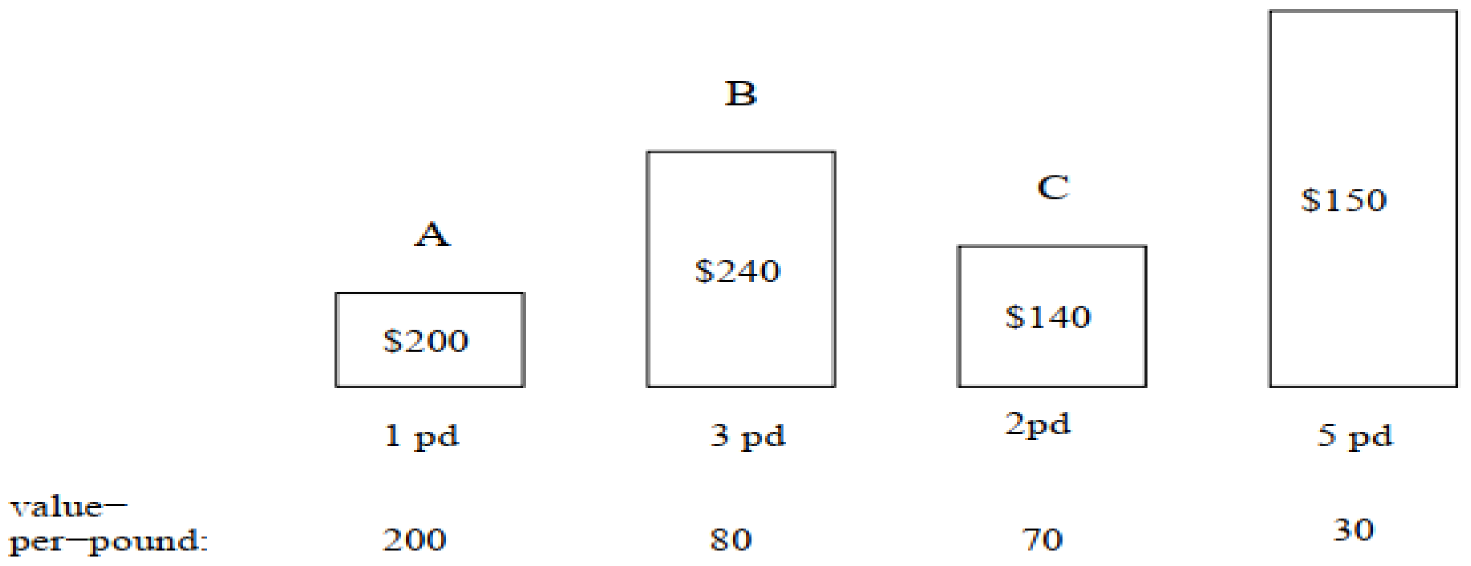
What should be the greedy choice?

1. Take the highest value item first.

2. Take the smallest weight item first.
   - So that you can carry more items. If you carry more items, you'll get more value/profit(?)

Greedy Solution

Sort items by decreasing value-per-pound



If knapsack holds $K = 5$ pd, solution is:

1	pd	A
3	pd	B
1	pd	C

Greedy Solution

- Calculate the value-per-pound $\rho_i = \frac{v_i}{w_i}$ for $i = 1, 2, \ldots, n$.
- Sort the items by decreasing ρ_i.
  Let the sorted item sequence be $1, 2, \ldots, i, \ldots n$, and the corresponding value-per-pound and weight be ρ_i and w_i respectively.
- Let k be the current weight limit (Initially, $k = K$).
  In each iteration, we choose item i from the head of the unselected list.
  - If $k \geq w_i$, set $x_i = 1$ (we take item i), and reduce $k = k - w_i$, then consider the next unselected item.
  - If $k < w_i$, set $x_i = k/w_i$ (we take a fraction k/w_i of item i), Then the algorithm terminates.

Running time: $O(n \log n)$.

Greedy Solution

- Observe that the algorithm may take a fraction of an item. This can only be the last selected item.
- We claim that the total value for this set of items is the optimal value.

Dry Run

Item i	1	2	3	4	5
Value v	6	10	18	15	7
Weight w	1	2	4	5	7
v/w	6	5	4.5	3	1

THE ITEMS ARE SORTED BY THEIR v/w values. (if its not already sorted, you have to do the sorting)

Let capacity K = 10

1. Add 1st item. x1 = 1. Remaining capacity = 9
2. Add 2nd item. x2 = 1. Remaining capacity = 7
3. Add 3rd item. x3 = 1. Remaining capacity = 3
4. Add 4th item. But you can't add the complete 4th item. We need 3/5 of the 4th item. Note that in this fraction, numerator is the remaining capacity and denominator is the total weight of the 4th item.

So, Solution set is:

X = {1 , 1, 1, 3/5, 0}

Proof of Correctness

- Given a set of n items $\{1, 2, \ldots, n\}$.
  - Assume items sorted by per-pound values: $\rho_1 \geq \rho_2 \geq \ldots \geq \rho_n$.
- Let the greedy solution be $G = (x_1, x_2, \ldots, x_n)$
  - x_i indicates fraction of item i taken.
- Consider any optimal solution $O = (y_1, y_2, \ldots, y_n)$
  - y_i indicates fraction of item i taken in O (for all i, $0 \leq y_i \leq 1$).
  - Knapsack must be full in both G and O:

$$\sum_{i=1}^{n} x_i w_i = \sum_{i=1}^{n} y_i w_i = K.$$

  - Consider the first item i where the two selections differ.
    By definition, solution G takes a greater amount of item i than solution O (because the greedy solution always takes as much as it can). Let $x = xi - yi$.

Proof of Correctness

Consider the following new solution O' constructed from O:

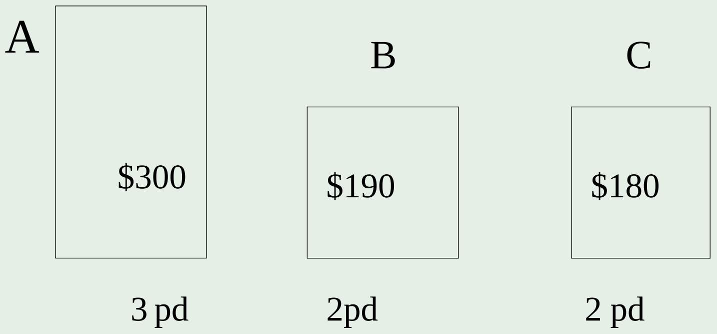
- For $j < i$, keep $y'_j = y_j$.
- Set $y'_i = x_i$.
- In O, remove items of total weight xw_i from items $i + 1$ to n, resetting the y'_j appropriately.

  This is always doable because $\sum_{j=i}^{n} x_j = \sum_{j=i}^{n} y_j$
- The total value of solution O' is greater than or equal to the total value of solution O (why?)
- Since O is largest possible solution and value of O' cannot be smaller than that of O, O and O' must be equal.
- Thus solution O' is also optimal.

By repeating this process, we will eventually convert O into G, without changing the total value of the selection.
Therefore G is also optimal!

Greedy solution for 0-1 Knapsack Problem?

The 0-1 Knapsack Problem does not have a greedy solution!



value–per–pound: 100 95 90

$K = 4$. Solution is item B + item C

Question

Suppose we tried to prove the greedy algorithm for 0-1 knapsack problem **does** construct an optimal solution. If we follow exactly the same argument as in the fractional knapsack problem where does the proof fail?

Slide Credits

- COMP 3711H Design and Analysis of Algorithms
  Fall 2014