Section 4.2: The $[0, 1]$-Knapsack Problem

Find a greedy algorithm to solve the following problem.

input:

- n objects such that object i ($1 \leq i \leq n$) has a positive *value* v_i and a positive *weight* w_i.
- A maximum weight W.

output: A vector $X = (x_1, x_2, ..., x_n)$ such that

- $0 \leq x_i \leq 1$
- The total value $x_1 v_1 + x_2 v_2 + ... + x_n v_n$ is maximized.
- The total weight is not too heavy: $x_1 w_1 + x_2 w_2 + ... + x_n w_n \leq W$.

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to values.

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to values.

- $x_3 = 1$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to values.

- $x_3 = 1$
- $x_5 = 1$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to values.

- $x_3 = 1$
- $x_5 = 1$
- $x_4 = 0.5$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to values.

- $x_3 = 1$
- $x_5 = 1$
- $x_4 = 0.5$
- Total value: 146

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to weights.

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to weights.

- $x_1 = 1$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to weights.

- $x_1 = 1$
- $x_2 = 1$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to weights.

- $x_1 = 1$
- $x_2 = 1$
- $x_3 = 1$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to weights.

- $x_1 = 1$
- $x_2 = 1$
- $x_3 = 1$
- $x_4 = 1$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50

Let us try to make greedy choices with respect to weights.

- $x_1 = 1$
- $x_2 = 1$
- $x_3 = 1$
- $x_4 = 1$
- Total value: 156

This proves that greedy choices with respect to values **does not** lead to an optimal solution in general.

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50
$\frac{v_i}{w_i}$	2	1.5	2.2	1	1.2

Let us try to make greedy choices with respect to values per unit of weight.

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50
$\frac{v_i}{w_i}$	2	1.5	2.2	1	1.2

Let us try to make greedy choices with respect to values per unit of weight.

- $x_3 = 1$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50
$\frac{v_i}{w_i}$	2	1.5	2.2	1	1.2

Let us try to make greedy choices with respect to values per unit of weight.

- $x_3 = 1$
- $x_1 = 1$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50
$\frac{v_i}{w_i}$	2	1.5	2.2	1	1.2

Let us try to make greedy choices with respect to values per unit of weight.

- $x_3 = 1$
- $x_1 = 1$
- $x_2 = 1$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50
$\frac{v_i}{w_i}$	2	1.5	2.2	1	1.2

Let us try to make greedy choices with respect to values per unit of weight.

- $x_3 = 1$
- $x_1 = 1$
- $x_2 = 1$
- $x_5 = 0.8$

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50
$\frac{v_i}{w_i}$	2	1.5	2.2	1	1.2

Let us try to make greedy choices with respect to values per unit of weight.

- $x_3 = 1$
- $x_1 = 1$
- $x_2 = 1$
- $x_5 = 0.8$
- Total value: 164

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50
$\frac{v_i}{w_i}$	2	1.5	2.2	1	1.2

Let us try to make greedy choices with respect to values per unit of weight.

- $x_3 = 1$
- $x_1 = 1$
- $x_2 = 1$
- $x_5 = 0.8$
- Total value: 164

This proves that greedy choices with respect to weights **does not** lead to an optimal solution in general.

Example: $W = 100$

i	1	2	3	4	5
v_i	20	30	66	40	60
w_i	10	20	30	40	50
$\frac{v_i}{w_i}$	2	1.5	2.2	1	1.2

Let us try to make greedy choices with respect to values per unit of weight.

- $x_3 = 1$
- $x_1 = 1$
- $x_2 = 1$
- $x_5 = 0.8$
- Total value: 164

This proves that greedy choices with respect to weights **does not** lead to an optimal solution in general.

What about greedy choices with respect to values per unit of weight.

Is this optimal ?!

Lemma

If the objects are selected in order of decreasing $\frac{v_i}{w_i}$, *then the greedy choice finds an optimal solution.*

Lemma

If the objects are selected in order of decreasing $\frac{v_i}{w_i}$, then the greedy choice finds an optimal solution.

PROOF:

Greedy choice with respect to value:

$x_3 = 1$
$x_5 = 1$ total value: 146
$x_4 = 0.5$

Greedy choice with respect to weight:

$x_1 = 1$
$x_2 = 1$ total value: 156
$x_3 = 1$
$x_4 = 1$

Greedy choice with respect to value per unit weight

$x_3 = 1$
$x_1 = 1$ total value: 164
$x_2 = 1$
$x_5 = 0.8$

Is this optimal?

<u>Lemma:</u> If the objects are selected in order of decreasing $\frac{v_i}{w_i}$, then the greedy choice finds an optimal solution.

<u>Proof:</u> We can suppose that

$$\frac{v_1}{w_1} \geq \frac{v_2}{w_2} \geq \cdots \geq \frac{v_n}{w_n}.$$

Let $X = (x_1, x_2, \ldots, x_n)$ be a solution corresponding to the greedy algorithm with respect to $\frac{v_i}{w_i}$.

If $x_i = 1$ for all $1 \le i \le n$, then the solution is optimal.

Otherwise, let $1 \le j \le n$ be the smallest index such that $x_j < 1$. We have

$$x_i = 1 \qquad (i < j)$$

$$0 \le x_j < 1$$

$$x_i = 0 \qquad (i > j)$$

$$\sum_{i=1}^{n} x_i w_i = W \qquad (1)$$

Let $Y = (y_1, y_2, \ldots, y_n)$ be any solution such that

$$\sum_{i=1}^{n} y_i w_i \le W \qquad (2)$$

We want to show that Y is not better than X.

and $0 \le y_i \le 1 \quad (1 \le i \le n)$

From (1) and (2), we have

$$\sum_{i=1}^{n} x_i w_i - \sum_{i=1}^{n} y_i w_i \geq 0$$

$$\sum_{i=1}^{n} (x_i - y_i) w_i \geq 0 \qquad (3)$$

Let us compare the value for X to the value for Y.

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

$$= \sum_{i=1}^{n} (x_i - y_i) v_i$$

$$= \sum_{i=1}^{n} (x_i - y_i) w_i \frac{v_i}{w_i}$$ We multiply by 1!

$$= \sum_{i=1}^{j-1} (x_i - y_i) w_i \frac{v_i}{w_i} + (x_j - y_j) w_j \frac{v_j}{w_j} + \sum_{i=j+1}^{n} (x_i - y_i) w_i \frac{v_i}{w_i}$$

For $1 \leq i < j$, $x_i = 1$.
So $x_i - y_i \geq 0$.
Moreover, $\frac{v_i}{w_i} \geq \frac{v_j}{w_j}$.
So $(x_i - y_i) w_i \frac{v_i}{w_i} \geq (x_i - y_i) w_i \frac{v_j}{w_j}$

For $j < i \leq n$, $x_i = 0$. So $x_i - y_i \leq 0$.
Moreover, $\frac{v_i}{w_i} \leq \frac{v_j}{w_j}$.
So $(x_i - y_i) w_i \frac{v_i}{w_i} \geq (x_i - y_i) \cdot w_i \cdot \frac{v_j}{w_j}$

$$\geq \sum_{i=1}^{j-1} (x_i - y_i) w_i \frac{v_j}{w_j} + (x_j - y_j) w_j \frac{v_j}{w_j} + \sum_{i=j+1}^{n} (x_i - y_i) w_i \frac{v_j}{w_j}$$

$$= \sum_{i=1}^{n} (x_i - y_i) w_i \frac{v_j}{w_j}$$

$$= \frac{v_j}{w_j} \sum_{i=1}^{n} (x_i - y_i) w_i$$

$$\geq 0 \qquad \text{by } (3).$$

So, whatever is Y, the value for Y is not better than the value for X.

□

<u>Conclusion</u>: We can solve this problem in $O(n \log(n))$ time:

Step 1: Compute all $\frac{v_i}{w_i}$'s. $O(n)$ time

Step 2: Sort the $\frac{v_i}{w_i}$'s $O(n \log(n))$ time

Step 3: Apply the greedy choice $O(n)$ time.

for a total of $O(n \log(n))$ time.

What do we do if x_i must satisfy $x_i \in \{0,1\}$ instead of $x_i \in [0,1]$? Refer to Chapter 5.

So here is how to solve the problem:

Step 1 : Compute all the $\frac{v_i}{w_i}$'s.

Step 2 : Sort the objects in decreasing order of $\frac{v_i}{w_i}$ (use Merge Sort).

Step 3 : Build the solution by applying the greedy choice with respect to decreasing order of $\frac{v_i}{w_i}$.

How much time does it take?

So here is how to solve the problem:

Step 1 : Compute all the $\frac{v_i}{w_i}$'s.

Step 2 : Sort the objects in decreasing order of $\frac{v_i}{w_i}$ (use Merge Sort).

Step 3 : Build the solution by applying the greedy choice with respect to decreasing order of $\frac{v_i}{w_i}$.

How much time does it take?

Step 1 : $O(n)$ time

Step 2 : $O(n\log(n))$ time

Step 3 : $O(n)$ time

Total time : $O(n) + O(n\log(n)) + O(n) = O(n\log(n))$

Is there a faster algorithm?

Is there a faster algorithm?

What do we do if x_i must satisfy $x_i \in \{0, 1\}$ instead of $x_i \in [0, 1]$?

Is there a faster algorithm?

What do we do if x_i must satisfy $x_i \in \{0, 1\}$ instead of $x_i \in [0, 1]$?

Refer to Chapter 5.