The Greedy Paradigm

"Construct a solution iteratively, via a sequence of myopic decision, and hope that everything work out int the end"

Fractional Knapsack problem

→ Given n objects and a knapsack with a capacity M (weight)

→ Each object has weight w_i and profit P_i

→ For each object i, a fraction $x_i (0 \le 1)$ can be placed in the knapsack, then the profit earned is $p_i x_i$

objective function is

Maximize $\sum_{i=1}^{n} p_i x_i$ --- (1)

Subject to $\sum_{i=1}^{n} w_i x_i \leq M$ --- (2)

where $0 \leq x_i \leq 1$ (3)

$p_i > 0$

$w_i > 0$

A feasible solution is any subset $\{x_1, x_2 \cdots x_n\}$ satisfying 2 & 3

An optimal solution is a feasible solution that maximize (1)

Let $n = 3$, $M = 20$

$(P_1, P_2, P_3) = (25, 24, 15)$

$(w_1, w_2, w_3) = (18, 15, 10)$

Some feasible solution.

		$\sum_{i=1}^{n} w_i x_i$	$\sum_{i=1}^{n} p_i x_i$
1-	$\frac{1}{2}, \frac{1}{3}, \frac{1}{4}$	16.5	24.25
2.	$1, 2/15, 0$	20	28.2
3	$(0, 1, 1/2)$	20	31.5

Strategy 1 :- Maximize objective function

Put the object with greatest profit in the knapsack

$$\sum_{i=1}^{n} p_i x_i = 25 \times 1 + 24 \times 2/15 + 0 = 28.5$$

$$\sum_{i=1}^{n} w_i x_i = 18 \times 1 + 15 \times 2/15 + 0 = 20$$

S 2 - Maximise capacity. (choose object with least weight)

$$\sum_{i=1}^{n} P_i x_i = 0 + 24 \times \frac{2}{3} + 15 \times 1 = 31$$

$$\sum_{i=1}^{n} w_i x_i = 0 + 15 \times \frac{2}{3} + 10 \times 1 = 20$$

S 3 - Balancing profit and capacity.

~~Put the object with the greatest~~

~~profit~~: Find the object by maximum profit per unit of capacity compute

$P_i/w_i = \frac{P_1}{w_1} \quad \frac{P_2}{w_2} \quad \frac{P_3}{w_3}$ (25, 24, 15) (18, 15, 10)

(1.38 1.6 1.5)

Start with largest.

$$\sum_{i=1}^{n} P_i x_i = P_2 x_2 + P_3 x_3 + P_1 x_1$$

$$= \boxed{1.6 \times 15 + 1.5 \times 5 + 1.38 \times 0 = 31.5}$$

Algorithm.

Iteratively picks the item with largest value-per-weight ratio $\left(\frac{p_i}{w_i}\right)$

If, at the end, the knapsack cannot fit the entire last item with greatest value-per-weight, we will take fraction of it to fill the knapsack.

Running time :- This algorithm takes $O(n \log n)$ time to sort the items by the ratio and another $O(n)$ time to traverse and pick.

$$O(n \log n) + O(n)$$

$$\approx O(n \log n)$$

Scheduling

Job scheduling with Deadline.

$n = 5$

Jobs	T_1	T_2	T_3	T_4	T_5
profit	20	15	10	5	1
Dead line	2	2	1	3	3

Each Job take unit time to complete
Let say 1 hour -

How many schedule possible

$n!$. but 0-1 way . Because it not possible to schedule every Job with deadline.
Objective is Maximise the profit.

Constraint is before deadline.

Solution:- Maximum dead line is 3 means

~~So~~ task T_4 and T_5 can wait for 2

hours -

So After 3 hour none of the Job

available.

Schedule would be done in only 3 slots.
To select the Job we find the maximum profit Job with dead line with in 3 hour.

0 —— 1 —— 2 —— 3
T_2 T_1 T_4

T_3 and T_5 can't schedule.

profit $= P(T_2) + P(T_1) + P(T_4)$

$= 20 + 15 + 5$

Other alternative schedule is

$(T_1 \; T_2 \; T_4)$

Job	Slot	Solution	Profit
T_1	1-2	T_1	20
T_2	0-1, 1-2	T_1, T_2	20+15
T_3	0-1, 1-2	T_1, T_2	20+15
T_4	0-1, 1-2, 2-3	T_1, T_2, T_4	20+15+5
T_5	"	"	40