Greedy Algorithm/Fractional Knapsack

Monday, January 11, 2021 1:44 PM

- Mostly used in ***Optimization problems***.

- Problems usually have n inputs and require to obtain a subset/ordering of inputs that satisfy some ***constraint**s*.
- Any subset that satisfies these constraints is called ***feasible solution***.
- Find a feasible solution that maximizes/minimizes an ***objective function***.
- A feasible solution that maximizes/minimizes an ***objective function*** is called an ***optimal solution***.

Greedy method usually works in ***stages***.

- At each stage an input is selected to be included in the optimal solution.
- The selection procedure is based on some ***optimization measure***.
- There may be different optimization measures for a given problem.
- The optimization measure may be the objective function.
- If the inclusion will cause the solution to be infeasible, it is discarded.

```
Algorithm Greedy(a, n)
// a[1 : n] contains the n inputs.
{
    solution := ∅; // Initialize the solution.
    for i := 1 to n do
    {
        x := Select(a);
        if Feasible(solution, x) then
            solution := Union(solution, x);
    }
    return solution;
}
```

FRACTIONAL KNAPSACK PROBLEM:

Given n objects and a knapsack/bag of some capacity. Object i has weight w_i and the knapsack has capacity m. If a fraction x_i, $0<=x_i<=1$, is placed into the knapsack, then a profit of $p_i x_i$ is earned.

The objective is to obtain a filling of the knapsack that maximizes the profit earned.

As the knapsack has capacity m, the sum of weights of objects included in the knapsack must be at most m.

(The profits and weights are positive numbers.)

Formally,

$$\text{maximize} \sum_{1 \le i \le n} p_i x_i \qquad (1)$$

$$\text{subject to} \sum_{1 \le i \le n} w_i x_i \le m \qquad (2)$$

$$\text{and } 0 \le x_i \le 1, \quad 1 \le i \le n \qquad (3)$$

A ***feasible solution*** (or filling) is any set (x_1, … x_n) that satisfies eqs. 2 & 3. An ***optimal solution*** is any feasible solution that maximize eq. 1.

$\frac{2}{15} \times 15$

Example:
n = 3,
m = 20.
(p_1,p_2,p_3) = (25, 24, 15)
(w_1,w_2,w_3) = (18, 15, 10)
=(1.38,1.6,1.5)

9+5+2.5   12.5+8+3.75

	(x_1,x_2,x_3)	$\boldsymbol{\Sigma w_i x_i}$	$\boldsymbol{\Sigma p_i x_i}$
1. Random <u>Profit</u>	(1/2,1/3,1/4)	16.5	24.25
2. Maximum weight	(1,2/15,0)	18+2	25*1+24*2/15 = 28.2
3. Minimum Weight	(0,2/3,1)	10+10	15*1+24*2/3 =15+16=31
4 . Profit/weight	(0,1,1/2)	15 + 5	24 + 15*1/2 =24+7.5=31.5