UNIT 4: Greedy Technique

Greedy Technique:

Fractional Knapsack Problem

The Knapsack problem is a class of optimization problems in which we have to find the maximal answer among all the various possibilities given in the question.

There are three types of knapsack problems i.e.

0-1 Knapsack,
Fractional Knapsack, and
Unbounded Knapsack

Fractional Knapsack Problem

- is a classic example of a problem that can be efficiently solved using a greedy algorithm.

- In the optimal solution for the fractional variant of the knapsack problem, taking fractions of items are allowed rather than entire items.

Problem Definition

In the fractional knapsack problem, we are given:

- A set of **n** items, each with a weight $\mathbf{w_i}$ and a value $\mathbf{v_i}$.
- A knapsack with a maximum weight capacity **W**.
- The objective is to maximize the total value of items in the knapsack without exceeding its weight capacity **W**.
- Unlike the 0/1 knapsack problem, we can take fractions of an item, meaning we can split the item if needed.

Mathematical Formulation

- Let $\mathbf{x_i}$ be the fraction of item **i** included in the knapsack, where $\mathbf{0 \leq x_i \leq 1}$.
- The objective is to maximize the total value:

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

- Subject to the constraint:

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

Greedy Method Approach

The greedy method for the fractional knapsack problem is based on selecting items with the highest value-to-weight ratio first.

This ensures that each unit of weight added to the knapsack contributes the maximum possible value.

Steps of the Greedy Algorithm:

1. **Calculate Value-to-Weight Ratio**: For each item **i**, calculate the value-to-weight ratio, $\mathbf{v_i/w_i}$
2. **Sort Items:** Sort all items in descending order of their value-to-weight ratio.
3. **Initialize Knapsack:** Start with an empty knapsack with a current weight of 0 and a total value of 0.
4. **Iterate Over Sorted Items:**
   - For each item, check if adding the entire item would exceed the knapsack's capacity.
   - If the entire item can be added without exceeding capacity, add it and update the total weight and value.
   - If only a fraction can be added, add as much as possible to fill the knapsack to its capacity.
5. **Stop When Knapsack is Full:** The algorithm stops when the knapsack reaches its weight capacity W.

Greedy Algorithm: Fractional Knapsack

ALGORITHM FractionalKnapsack
//Fractional knapsack solution using Greedy Technique
//Input: List of items with their weights (w[i]) and values (v[i]) ,
// Maximum weight capacity of knapsack W
//Output: Maximum total value of items in the knapsack

1. ratio[i] = v[i] / w[i] //Calculate the value-to-weight ratio for each item
2. Sort the items based on ratio[i] in descending order
3. totalValue = 0, currentWeight = 0
4. For each item i in the sorted list:
   If currentWeight + w[i] ≤ W // Add the entire item to the knapsack
   currentWeight += w[i]
   totalValue += v[i]
   Else // Add fraction of the item to fill the knapsack
   fraction = (W - currentWeight) / w[i]
   totalValue += v[i] * fraction
   currentWeight = W (Knapsack is full)
5. Return totalValue

Apply Greedy strategy to solve the following instance of fractional knapsack problem.

Item	Weight (w_i)	Value (v_i)
1	15	100
2	10	60
3	25	120
4	30	90
5	5	50

$$W = 50,$$

Solution:

STEP 1. Calculate Value-to-Weight Ratios

Item	Weight (w_i)	Value (v_i)	Value-to-Weight Ratio ($\frac{v_i}{w_i}$)
1	15	100	6.67
2	10	60	6.00
3	25	120	4.80
4	30	90	3.00
5	5	50	10.00

Objective:
Maximize the total value of items in the knapsack without exceeding the weight capacity W=50.

STEP 2. Sort Items by Value-to-Weight Ratio:

Sort the items in descending order of their value-to-weight ratio

Item 5: $\frac{50}{5} = 10.00$

Item 1: $\frac{100}{15} = 6.67$

Item 2: $\frac{60}{10} = 6.00$

Item 3: $\frac{120}{25} = 4.80$

Item 4: $\frac{90}{30} = 3.00$

STEP 3: Initialize Variables:

Total weight in the knapsack: currentWeight = 0

Total value in the knapsack: totalValue = 0

STEP 4. Iterate Over Sorted Items:
Add items to the knapsack in the order of their value-to-weight ratio, until the knapsack is full.

Item 5: Weight = 5, Value = 50

Current Weight = 0 + 5 = 5

Total Value = 0 + 50 = 50

Item 1: Weight = 15, Value = 100

Current Weight = 5 + 15 = 20

Total Value = 50 + 100 = 150

Item 2: Weight = 10, Value = 60

Current Weight = 20 + 10 = 30

Total Value = 150 + 60 = 210

Item 3: Weight = 25, Value = 120

*If we add the entire item, the total weight would be 30+25=55, which exceeds the knapsack capacity W=50****, so we can only add a fraction of this item***

Available remaining capacity in the knapsack: 50−30=20
Fraction of Item 3 to add: 20/25=0.8
Value contributed by the fraction of Item 3: 120×0.8=96
Current Weight: 50 (knapsack full)
Total Value: 210+96=306

Item 4: Not considered because the knapsack is already full.

Final values:

Current Weight = 50 (knapsack full)
Total Value = 210 + 96 = 306

STEP 5. Stop When Knapsack is Full:

The knapsack is now full with a total weight of 50 and a total value of 306.

Summary of the selection

Item	Fraction Added	Weight Added	Value Added	Total Weight	Total Value
5	1 (whole)	5	50	5	50
1	1 (whole)	15	100	20	150
2	1 (whole)	10	60	30	210
3	0.8	20	96	50	306

optimal solution using greedy method for the given fractional knapsack problem is:

- Fully include Item 5, Item 1, and Item 2.
- Include 80% (fraction) of Item 3.

Total value in the knapsack: 306
Total weight in the knapsack: 50

Let’s check our understanding

Item	Weight (w_i)	Value (v_i)
1	10	60
2	20	100
3	30	120

$$W = 50,$$

Solve the above instance of Fractional Knapsack problem using Greedy Strategy.

Let’s check our understanding

Knapsack Capacity: $W = 60$

Item	Weight (w_i)	Value (v_i)
1	20	100
2	30	120
3	10	60
4	40	240
5	15	90

Solve the above instance of Fractional Knapsack problem using Greedy Strategy.

Optimal Solution:

- **Total Weight:** 60
- **Total Value:** 60 (Item 3) + 240 (Item 4) + 60 (fraction of Item 5) = **360**

Let’s check our understanding

Knapsack Capacity: $W = 100$

Item	Weight (w_i)	Value (v_i)
1	25	120
2	50	200
3	30	180
4	10	40
5	5	25
6	60	300

Solve the above instance of Fractional Knapsack problem using Greedy Strategy.

Optimal Solution:

- **Total Weight:** 95
- **Total Value:** 180 (Item 3) + 300 (Item 6) + 50 (fraction of Item 5) = **530**

Let’s check our understanding

Knapsack Capacity: $W = 75$

Item	Weight (w_i)	Value (v_i)
1	10	50
2	20	60
3	30	120
4	40	200
5	50	150
6	60	180
7	5	45

Optimal Solution:
- **Total Weight:** 75
- **Total Value:** 45 (Item 7) + 50 (Item 1) + 200 (Item 4) + 80 (fraction of Item 3) = **375**

Solve the above instance of Fractional Knapsack problem using Greedy Strategy.

Time Complexity

- **Sorting:** The dominant factor is sorting the items by their value-to-weight ratio, which takes **O(n log n)** time.

- **Item Selection:** Iterating through the sorted list takes O(n) time.

- **Total Complexity: O(n log n)**

Advantages

- **Optimal Solution:** The greedy approach guarantees an optimal solution for the fractional knapsack problem.
- **Efficiency:** The algorithm is efficient with a time complexity of O(n log n).
- **Simplicity:** It is easy to understand and implement.

Limitations

- **Not Suitable for 0/1 Knapsack**: The greedy method does not guarantee an optimal solution for the 0/1 knapsack problem (where items cannot be split).

- **Assumes Divisibility:** The method works under the assumption that items can be divided into fractions.

Applications

- Resource Allocation Problems: Optimally allocate resources when items can be divided.
- Investment Decisions: Allocate a limited budget to investments with fractional returns.
- Load Balancing: Distribute tasks or loads in a fractional manner for optimization.

Resources

- https://www.youtube.com/watch?v=F_DDzYnxO14
- https://algorithm-visualizer.org/