Heaven's Light is Our Guide

Computer Science & Engineering
Rajshahi University of Engineering & Technology

Lab Manual

Module-06

Course Title : Sessional based on CSE 2201
Course No. : CSE 2202

Experiment No. 6

Name of the Experiment: Design and Complexity analysis of Greedy Methods

Date: 6th Cycle

Algorithms (Greedy Method)**:**

- **Knapsack Problem**
- **Tree Vertex Splitting**
- **Job Sequence with Deadline**
- **Minimum-cost Spanning Tree (Prom's and Kruskal's Algorithm)**
- **Single-source Shortest Path**

Knapsack Problem:

Now, let us try to apply the greedy method to solve a more complex problem. This problem is the knapsack problem. We are given n objects and a knapsack. Object i has a weight w, and the knapsack has a capacity M. If a fraction xit 0 < x,< < 1, of object i is placed into the knapsack then a profit of piXi is earned. The objective is to obtain a filling of the knapsack that maximizes the total profit earned. Since the knapsack capacity is M, we require the total weight of all chosen objects to be at most M. Formally, the problem may be stated as:

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

$$\text{subject to} \sum_{1 \le i \le n} w_i x_i \le M$$

$$\text{and} \quad 0 \le x_i \le 1, \quad 1 \le i \le n$$

The profits and weights are positive numbers.

A feasible solution (or filling) is any set (x_1, ..., x_n) satisfying above equations. An optimal solution is a feasible solution for which profit is maximized.

```
void greedy_knapsack ( m, n )
{
    // Solution vector is x[i], 0 <= i < n

    for ( i = 0; i < n; i++ )
        x[i] = 0.0;

    U = m;                              // Unused capacity
    for ( i = 0; ( i < n ) && ( w[i] <= U ); i++ )
    {
        x[i] = 1.0;
        U = U - w[i];
    }
    if ( i < n )
        x[i] = U / w[i];
}
```

Tree Vertex Splitting:

Let T = (V,E,w) be a weighted directed tree, V is the set of vertices, E is the set of edges, w is the weight function for the edges, w_{ij} is the weight of the edge <i, j> ∈ E. We say that w_{ij} = ∞ if <i, j> ∉ E. ∗ A vertex with in-degree zero is called a source vertex. A vertex with out-

degree zero is called a sink vertex. For any path P ∈ T, its delay d(P) is defined to be the sum of the weights (w_{ij}) of that path, or

$$d(P) = \sum_{\langle i,j \rangle \in P} w_{ij}$$

Delay of the tree T, d(T) is the maximum of all path delays.
The objective of this problem solution is to be Splitting vertices to create a forest. Let T/X be the forest that results when each vertex u ∈ X is split into two nodes u^i and u^o such that all the edges <u, j> ∈ E [<j, u^i >∈ E] are replaced by edges of the form <u^o, j> ∈ E [<j, u^i> ∈ E]. Outbound edges from u now leave from u^o. Inbound edges to u now enter at u^i. Split node is the booster station.

- ✓ Tree vertex splitting problem is to identify a set X ⊆ V of minimum cardinality (minimum number of booster stations) for which d(T/X) ≤ δ for some specified tolerance limit δ
  - ➢ TVSP has a solution only if the maximum edge weight is ≤ δ

- ✓ Given a weighted tree T = (V,E,w) and a tolerance limit δ, any X ⊆ V is a feasible solution if d(T/X) ≤ δ
  - ➢ Given an X, we can compute d(T/X) in O(|V |) time
  - ➢ A trivial way of solving TVSP is to compute d(T/X) for every X ⊆ V , leading to a possible 2|V | computations

Task:

1. Find out the complexity of the above algorithms.
2. Code the above algorithm in any language(i.e. C/C++/Java)
3. Find the running time for a set of points list (let size 1000, 5000,10000, 15000 etc.
4. Write down a report on it.

Recommended Exercise:
Programming Exercises of Chapter 4: "Greedy Method" of "Fundamentals of Computer Algorithm", Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran.