Homework

Nguyen Pham Phuong Nam - 23520978
Pham Huynh Long Vu - 23521813

21 November 2024

Approximation Algorithms for Set Cover and Travelling Salesman Problem

1. Set Cover Problem

Problem Description

Given a set U, which is a collection of elements, and a family of subsets $S = \{S_1, S_2, \ldots, S_m\}$ of U, where each subset $S_i \subseteq U$, the objective is to select the minimum number of subsets such that every element in U is covered at least once.

Formal Definition:

Given:

$$U = \{u_1, u_2, \ldots, u_n\}, \quad S = \{S_1, S_2, \ldots, S_m\}, \quad S_i \subseteq U \quad \forall i.$$

The goal is to find a subset $S' \subseteq S$ such that:

$$\bigcup_{S_i \in S'} S_i = U,$$

and minimize the size of S':

$$\min |S'|.$$

Approximation Algorithms

Method 1: Greedy Algorithm

Idea: At each step, select the subset S_i that covers the largest number of uncovered elements. Repeat until all elements in U are covered.

Algorithm:

1. Initialize:
   - Set of uncovered elements $C \leftarrow U$.
   - Result set $S' \leftarrow \emptyset$.
2. While $C \neq \emptyset$, repeat:
   - Select $S_i \in S$ such that $|S_i \cap C|$ is maximized.
   - Add S_i to S'.
   - Update $C \leftarrow C \setminus S_i$ (remove covered elements from C).
3. Return S' as the set of selected subsets.

Complexity: $O(mn^2)$, where $n = |U|$ and $m = |S|$.

Approximation Ratio: The greedy algorithm achieves an approximation ratio of $H(d)$, where d is the size of the largest subset, and $H(d)$ is the d-th harmonic number:

$$H(d) = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{d}.$$

Method 2: Linear Programming (LP) Rounding

Idea: Formulate the problem as a linear program (LP) and round the fractional solution to obtain an integer solution.

Formulate the LP:

$$Minimize \sum_{i=1}^{m} x_i,$$

subject to:

$$\sum_{i:u \in S_i} x_i \geq 1 \quad \forall u \in U, \quad x_i \in [0, 1].$$

Rounding: Select S_i if $x_i \geq \alpha$ (where α is an appropriate threshold).

Complexity: Solving the LP takes $O(m^3)$, and rounding takes $O(mn)$.

2. Travelling Salesman Problem (TSP)

Problem Description

The Travelling Salesman Problem (TSP) aims to find a Hamiltonian cycle in a graph such that:

- Each vertex is visited exactly once.
- The total cost of the cycle is minimized.

Formal Definition:

Given a graph $G = (V, E)$:

- $V = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices.
- E is the set of edges with weights $c(v_i, v_j)$ representing cost, time, or distance.

Find a Hamiltonian cycle such that the total weight is minimized.

Approximation Algorithms

Method 1: Greedy Algorithm

Idea: Start from an arbitrary vertex. At each step, choose the smallest edge connecting the current vertex to an unvisited vertex.

Algorithm:

1. Select a starting vertex v_1.
2. While there are unvisited vertices:
   - Choose the smallest edge (v_i, v_j) where v_j is unvisited.
3. Return to the starting vertex to complete the cycle.

Complexity: $O(n^2)$, where $n = |V|$.

Method 2: 2-Approximation Algorithm (Minimum Spanning Tree - MST)

Idea: Build a Minimum Spanning Tree (MST) and perform an Euler Tour to construct a Hamiltonian cycle.

Algorithm:

1. Construct the MST of G using Kruskal's or Prim's algorithm.
2. Perform an Euler Tour on the MST to visit all vertices.
3. Remove duplicate visits, keeping the first occurrence of each vertex.

Complexity: $O(n^2)$.

Approximation Ratio: The cycle's total weight is at most twice the optimal solution.