Jinuk Hong

The Postman Tour: Definition, Algorithm, and Applications

1. Introduction

The Postman Tour, also known as the Chinese Postman Problem, is a classic problem in graph theory, closely related to the more well-known Traveling Salesman Problem (TSP). While TSP focuses on finding the shortest possible route that visits each vertex exactly once, the Postman Tour deals with visiting each edge at least once in the most efficient manner possible. This problem is particularly relevant in practical applications such as mail delivery, street cleaning, and other logistics where every street (or edge) must be traversed. This report delves into the definition of the Postman Tour, explores the algorithms used to solve it, and discusses its practical applications.

2. Definition and Problem Statement

2.1. Postman Tour

A Postman Tour on a connected and finite graph is a closed walk that uses each edge of the graph at least once. If the graph has edge weights, an optimal Postman Tour is one that has the minimum total edge-weight. In cases where the graph does not have specific weights, each edge can be assigned a weight of 1, making the optimal Postman Tour correspond to the shortest possible tour.

> Definition 6.3 (Postman Tour): A postman tour on a connected and finite graph is a closed walk that uses each edge of the graph at least once. If the graph is endowed with edge weights, an optimal postman tour is a postman tour with the minimum total edge-weight.

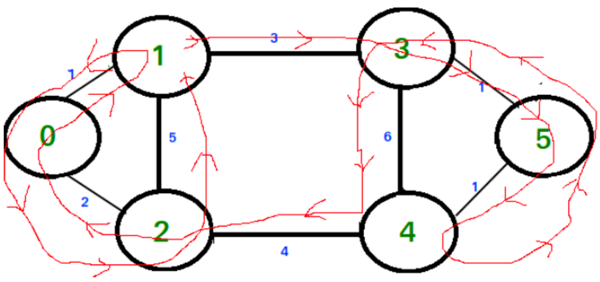

Fig1. Optimal postman tour for a weighted graph

2.2. Mathematical Formulation

The Postman Tour can be formulated mathematically as follows:

Let $G = (V,E)$ be a connected, undirected graph where V represents the set of vertices and E represents the set of edges, each with an associated weight $w(e)$ for $e \in E$

1) Objective: Find a closed walk W that starts and ends at the same vertex, visiting

each edge at least once, and minimizing the total weight.

2) Variable:

- x_{uv}: Binary variable indicating whether edge (u, v) is used in the walk.

3) Constraints:

- Each edge (u,v)∈E must be visited at least once.

$$\sum_{(u,v)\in E} x_{uv} \geq 1 \quad \forall (u, v) \in E$$

- The solution must form a closed walk.

$$\sum_{u\in V} x_{uv} = \sum_{u\in V} x_{vu} \quad \forall v \in V$$

  This equality ensures that the number of times each vertex is entered equals the number of times it is exited. This balanced flow is essential for forming a closed walk, as it allows the walk to start and end at the same vertex while ensuring that each edge is traversed at least once.

4) Objective Function:

$$minimize \sum_{(u,v)\in E} w_{uv} x_{uv}$$

3. Algorithm for the Postman Tour

3.1. Problem Definition and Solution Strategy

To solve the Postman Problem, especially for graphs with vertices of odd degrees, an algorithm is used that artificially adds some weighted edges between these vertices. The goal is to keep the additional weights to a minimum. Once these edges are added, the graph becomes Eulerian, allowing any Euler tour to be chosen as the optimal Postman Tour.

The idea behind the algorithm is to pair up the vertices with odd degrees in such a way that the sum of the weights of the edges added to make these pairs is minimized. This transforms the graph into one where every vertex has an even degree, making it Eulerian.

3.2. Algorithm 6.5 for an Undirected Graph:

1) Identify all vertices with odd degrees.
2) Compute the shortest paths between all pairs of these odd-degree vertices.
3) Find the minimum weight matching of these vertices based on the shortest paths computed.

4) Add the corresponding edges to the graph.
5) Find an Eulerian circuit in the modified graph, which corresponds to the optimal Postman Tour.

For directed graphs, the process is slightly more complicated but follows similar principles.

4. Problem Definition and Solution Strategy

The Postman Tour problem seeks the shortest route that starts and ends at the same vertex while traversing each edge at least once. For an undirected, connected graph G=(VE) with distances $d: E \rightarrow R^{+}$, the goal is to find this optimal tour.

4.1.Understanding the Problem

If G is Eulerian (every vertex has an even degree), a closed walk exists that visits each edge exactly once. This walk is optimal as it covers each edge the minimum number of times required. However, if some vertices have odd degrees, the graph isn't Eulerian, and some edges must be revisited. The challenge is to find a tour minimizing the total distance while ensuring all edges are covered at least once.

4.2.Example:

Consider an example graph G where some vertices have odd degrees. The goal is to find the shortest closed walk that covers all edges at least once and returns to the starting point.

1) T-join and Minimum-cost Perfect Matching:

To solve this problem, we introduce the concept of a T-join. A T-join in a graph is a subset of edges such that the vertices in a specified subset T have odd degrees, and all other vertices have even degrees. The subset T must contain an even number of vertices for a T-join to exist.

2) Definition of T-join

Definition 1 (T-join): Given an undirected graph

G=(V,E) and a subset of vertices T⊆V with even cardinality, a T-join is a subset J⊆E such that vertices in T have odd degrees in J, and vertices not in T have even degrees in J.

3) Algorithm for Minimum T-join

To find the minimum T-join, we use the concept of minimum-cost perfect matching. The algorithm can be summarized as follows:

3-1) Construct an auxiliary graph H with vertices corresponding to T and edges representing the shortest paths between vertices in T.

3-2) Find the minimum-cost perfect matching in H.

3-3) Translate the matching back to the original graph G to form the minimum T-join.

4) Algorithm 1: An Algorithm for the Minimum T-join

- Input: An undirected graph G=(V,E), distances $d: E \rightarrow Q^{+}$, T⊆V with |T| even.
- Output: The minimum T-join.

Steps:

1. Construct a complete graph H with vertex set T.
2. Define edge weights c(u,v) in H as the shortest path distances between u and v in G.
3. Find the minimum-cost perfect matching M in H.
4. If $c(M) = +\infty$, return that no T-join exists.
5. Initialize $J \leftarrow \emptyset$.
6. For each edge $(u,v) \in M$:
   - Find the shortest path P between u and v in G.
   - Update J as the symmetric difference of J and P.
7. Return J.

This algorithm effectively finds the minimum T-join by solving the minimum-cost perfect matching problem on the auxiliary graph H and then mapping the solution back to the original graph G.

5. Additional Insights: Graph Theory and the Postman Problem

Graph theory is not only about simplifying complex relationships into visually intuitive models but also about finding optimal solutions and efficient routes for various problems. For instance, if a postman needs to deliver mail to every house in a neighborhood, the optimal route would be one that passes through every street (edge) at least once. This problem, known as the Chinese Postman Problem, seeks the shortest closed walk that traverses every edge of the graph at least once.

To solve the Postman Problem effectively, one needs more than just a graph showing connections; the distances or weights of each edge must also be known. A weighted graph, where each edge has an associated cost or distance, allows the use of suitable algorithms to find the optimal route.

For example, consider driving from Nagoya to Kyoto. The shortest path can be found by representing intersections, start, and end points as vertices, and roads as edges in a graph. By assigning weights to each edge based on travel time or distance, the shortest route can be determined using appropriate algorithms.

5.1.Max-Flow Min-Cut Theorem in Transportation Networks

Futhermore, I thought we can implement maximum flow algorithm from chapter 9 into this situation. When transporting maximum goods from Nagoya to Kyoto, the routes can be represented as a graph where each edge has a capacity. The total flow of goods is limited by the capacities of these edges. The Max-Flow Min-Cut Theorem states that the maximum flow through a network equals the total weight of the edges in the smallest cut that separates the source from the sink. This theorem, derived from studying ways to disrupt supply lines during warfare, illustrates that cutting the most critical, low-capacity routes will significantly impact the maximum flow.

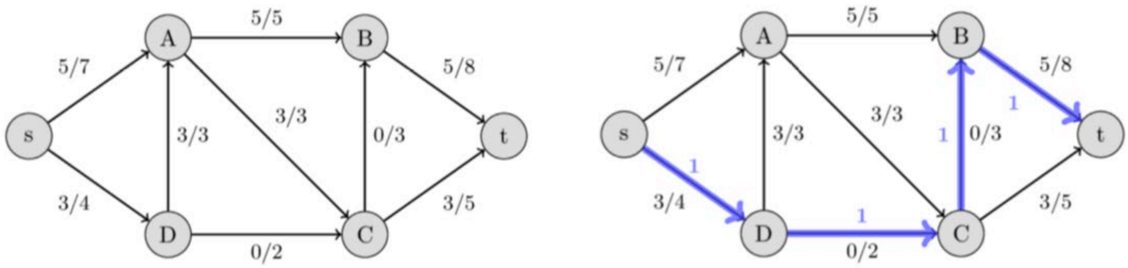

Fig2. Flow increasing through one path

6. Conclusion

The Postman Tour is a crucial problem in graph theory with significant practical applications in various fields. By understanding the principles behind the Postman Tour and the algorithms used to solve it, one can devise efficient solutions to real-world problems involving traversal of all edges in a graph. The algorithm to solve the Postman Tour, particularly for graphs with vertices of odd degrees, involves transforming the graph into an Eulerian one by adding minimal weight edges, thus enabling the use of an Eulerian circuit as the optimal solution. This problem's relevance extends to urban logistics, network maintenance, and robotics, highlighting its importance in both theoretical and applied contexts. Furthermore, understanding graph theory concepts like weighted graphs and the Max-Flow Min-Cut Theorem provides deeper insights into optimizing complex networks and routes.

7. References

1) "Graph Theory and Its Applications" - Relevant sections on Eulerian circuits and the Postman Problem.

2) Hwang, J., & Kim, Y. (2022). Greedy-based Neighbor Generation Methods of Local Search for the Traveling Salesman Problem. Journal of The Korea Society of Computer and Information, 27(9), 69-76. https://doi.org/10.9708/jksci.2022.27.09.069

3) Tiwari, S. P., Kumar, S., & Bansal, K. K. (2014). A Survey of Metaheuristic Algorithms for Travelling Salesman Problem. International Journal Of Engineering Research & Management Technology, 1(5).

4) Palhares, R. A., & Araujo, M. C. B. (2018). Vehicle Routing: Application of Travelling Salesman Problem in a Dairy. IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 1421–1425.

5) Nilsson, C. (2003). Heuristics for the Traveling Salesman Problem. Technical Report, Linköping University, Sweden.

6) 카이스트신문. 그래프 이론과 우체부 문제. Retrieved from http://times.kaist.ac.kr.