# Solving the Traveling Salesman Problem Using Quantum Annealing

Abid Jeem

**Abstract**

This paper explores how quantum annealing can be used to solve the Traveling Salesman Problem (TSP), a fundamental challenge in combinatorial optimization. Using a small-scale example with four cities, we implement a Quadratic Unconstrained Binary Optimization (QUBO) formulation and solve it using D-Wave's quantum annealing hardware. The results are compared to classical solutions, demonstrating the potential and limitations of current quantum approaches.

## 1 Introduction

- Background on combinatorial optimization and the TSP.
- Motivation for exploring quantum methods for hard problems.
- Overview of quantum annealing and D-Wave's approach.

## 2 Theoretical Background

### 2.1 The Traveling Salesman Problem

- Definition and formulation.
- Computational complexity and brute-force solution.

### 2.2 Quantum Annealing

- Description of adiabatic evolution.
- Advantages over classical methods for certain problems.

### 2.3 QUBO Formulation

- Definition and mathematical structure of QUBO.
- Mapping the TSP to a QUBO problem.
- Penalty terms for encoding constraints.

# 3 Implementation

## 3.1 Problem Setup

- The 4-city example and distance matrix.

## 3.2 Classical Baseline

- Exact or simulated annealing solution for comparison.

## 3.3 D-Wave Implementation

- Encoding the problem in Python using Ocean SDK.
- Embedding and sampling from the quantum annealer.

# 4 Results

- Output routes and energies from D-Wave.
- Comparison to the classical optimal solution.
- Visualization (histogram, route diagram).

# 5 Discussion

- Interpretation of D-Wave's results.
- Challenges: noise, constraint balancing, embedding.
- Scalability considerations and hybrid options.

# 6 Conclusion

- Summary of findings and insights.
- Reflection on quantum annealing's current capabilities.
- Future research directions.

# References


- D-Wave Systems documentation.
- Papers on QUBO, adiabatic theorem, and TSP.
- Python libraries used (Ocean SDK, NumPy, etc.).