Quantum vs. Classical Computing for the Traveling Salesman Problem: A Comparative Analysis Using IBM Qiskit

Walter Rodrigo Lopez

May 2025

1 Introduction

The Traveling Salesman Problem (TSP) requires finding the shortest route visiting all cities exactly once before returning to the origin. As an NP-hard problem with applications in logistics and network design, TSP's computational requirements grow factorially with problem size, making efficient solutions for large instances elusive.

Classical approaches include exact methods like brute force ($O(n!)$ complexity) that guarantee optimality but scale poorly, and heuristics like Nearest Neighbor ($O(n^2)$ complexity) that sacrifice optimality for efficiency. Quantum computing offers a potentially disruptive approach through its ability to represent multiple solutions simultaneously using superposition and entanglement, potentially exploring exponential solution spaces more efficiently.

The Quantum Approximate Optimization Algorithm (QAOA) leverages these quantum principles within a hybrid framework, using parameterized circuits to explore solution spaces while employing classical optimization to refine parameters. This research compares brute force and Nearest Neighbor algorithms against a QAOA implementation using Qiskit across small TSP instances. By evaluating solution quality, execution time, and optimality rates, we establish benchmarks for quantum methods and identify when they might achieve practical advantages as hardware capabilities advance.

2 Methodology

2.1 Problem Formulation

We generated random symmetric TSP instances with 5 cities distributed uniformly in a unit square $[0, 1]^2$. Distances between cities were calculated using the Euclidean metric. To ensure fair comparison, all algorithms were evaluated on identical problem instances across 20 trials with varying random seeds.

2.2 Classical Algorithms

We implemented two classical approaches representing different trade-offs between optimality and efficiency:

2.2.1 Brute Force Algorithm

The brute force approach examines all possible tours to guarantee optimal solutions:

Algorithm 1 Brute Force TSP Solver

Fix city 0 as the starting point
for each permutation p of cities $\{1, 2, \ldots, n-1\}$ **do**
  $tour \leftarrow [0] + p$
  $cost \leftarrow \text{CalculateTourCost}(tour)$
  if $cost < best\_cost$ **then**
    $best\_cost \leftarrow cost$
    $best\_tour \leftarrow tour$
  end if
end for
return $best\_tour, best\_cost$ =0

This algorithm has $O((n-1)!)$ time complexity, making it impractical for large instances but suitable as an optimality benchmark for our 5-city test cases.

2.2.2 Nearest Neighbor Algorithm

The Nearest Neighbor algorithm employs a greedy heuristic with $O(n^2)$ time complexity:

Algorithm 2 Nearest Neighbor TSP Solver

$current\_city \leftarrow 0$
$tour \leftarrow [current\_city]$
$unvisited \leftarrow \{1, 2, \ldots, n-1\}$
while $unvisited \neq \emptyset$ **do**
  $next\_city \leftarrow \arg\min_{c \in unvisited} distance(current\_city, c)$
  $tour \leftarrow tour + [next\_city]$
  $unvisited \leftarrow unvisited \setminus \{next\_city\}$
  $current\_city \leftarrow next\_city$
end while
return $tour, \text{CalculateTourCost}(tour)$ =0

While computationally efficient, this approach does not guarantee optimal solutions, especially when the optimal tour requires temporarily moving away from nearby cities.

2.3 Quantum Approach

We implemented a quantum solution based on the Quantum Approximate Optimization Algorithm (QAOA) framework using Qiskit.

2.3.1 Binary Encoding Strategy

Our approach encodes each city's position in the tour using binary representation:

- Each city requires $\lceil \log_2(n) \rceil$ qubits to represent its position in the tour
- For our 5-city problem, we used 3 qubits per city, totaling 15 qubits
- Bit values map directly to position indices (modulo n) in the tour

2.3.2 Parameterized Quantum Circuit

Our QAOA implementation used a parameterized circuit with the following structure:

Algorithm 3 QAOA Circuit for TSP

Apply H gates to all qubits (superposition)
for layer $l = 1$ to 2 **do**
    for qubit $q = 0$ to $n_{qubits} - 1$ **do**
        Apply $R_y(\theta_{l,q})$ to qubit q {Rotation layer}
    end for
    for qubit $q = 0$ to $n_{qubits} - 2$ **do**
        Apply CNOT between qubits q and $q + 1$ {Entanglement}
    end for
    Apply CNOT between qubit $n_{qubits} - 1$ and 0 {Circular entanglement}
end for
Apply final R_y rotations to all qubits
Measure all qubits =0

This circuit creates a superposition of potential solutions, with parameterized rotations determining the probability amplitude of each state. The entanglement operations establish correlations between qubits that can encode problem constraints.

2.3.3 Measurement Interpretation

A key challenge in our quantum approach was translating measurement outcomes into valid TSP tours:

This interpretation handles quantum uncertainty by prioritizing non-conflicting assignments and optimally resolving conflicts to ensure valid tours.

Algorithm 4 Quantum Measurement Interpretation

```
for each city i do
    Extract corresponding bits from measurement
    Convert to integer position: pos_i = int(bits_i) mod n
end for
Resolve position conflicts (multiple cities assigned to same position)
Fill vacant positions with unassigned cities
return ordered city sequence and tour cost =0
```

2.3.4 Parameter Optimization

We used the COBYLA classical optimizer to tune the quantum circuit parameters:

Algorithm 5 QAOA Optimization Process

```
Initialize random parameters θ⃗
Define objective function that returns average tour cost across 1024 shots
Optimize parameters with COBYLA (50 iterations)
Execute final circuit with optimized parameters
Return best tour found from measurement results =0
```

This hybrid quantum-classical approach leverages quantum parallelism to explore the solution space while using classical optimization to refine the search direction.

2.4 Experimental Setup

All experiments were conducted with the following configuration:

- 20 random 5-city TSP instances
- Qiskit AerSimulator for quantum circuit simulation
- 1024 shots per quantum circuit execution
- 50 iterations of COBYLA optimization

3 Results

We evaluated three approaches to solving the Traveling Salesman Problem—brute force, Nearest Neighbor, and QAOA—across 20 random 5-city instances. Our comparative analysis focused on three key metrics: execution time, solution quality, and optimality rate.

3.1 Execution Time

Figure 1 presents the average execution time for each algorithm on a logarithmic scale. The quantum approach (QAOA) exhibited substantially longer execution times, requiring an average of 10.76 seconds per instance—approximately 150,000 times slower than brute force (0.000072 seconds) and 538,000 times slower than Nearest Neighbor (0.000020 seconds). This dramatic difference in computational efficiency highlights the significant overhead currently associated with quantum simulation.

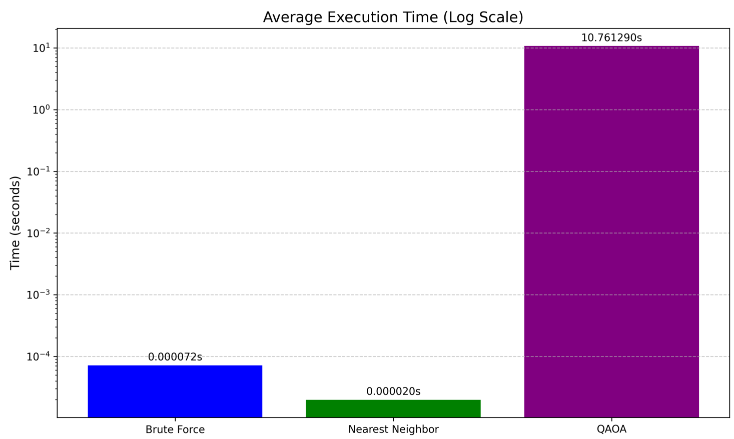

Figure 1: Comparison of average execution times (logarithmic scale)

3.2 Solution Quality

Despite vast differences in execution time, both brute force and QAOA consistently found tours of identical quality, with an average tour cost of 2.1219 (Figure 2). Nearest Neighbor, while significantly faster, produced slightly longer tours with an average cost of 2.2025, approximately 3.8% higher than the optimal solutions.

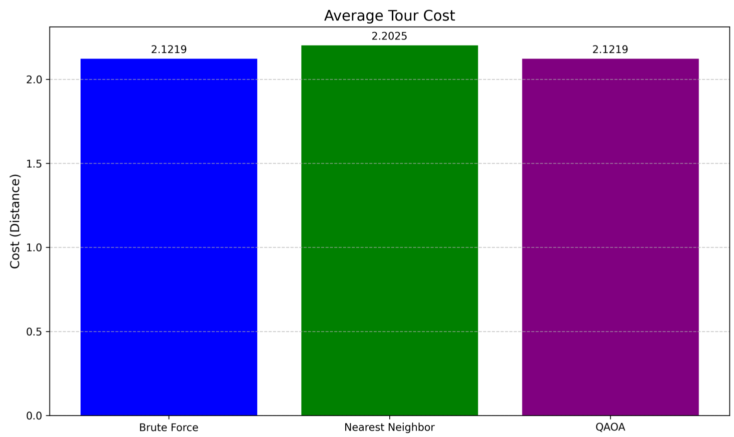

Figure 2: Comparison of average tour costs across algorithms

3.3 Optimality Rate

The percentage of trials in which each algorithm found the optimal solution is presented in Figure 3. Both brute force and QAOA achieved 100% optimality across all test instances, while Nearest Neighbor found optimal solutions in only 35% of trials. This result is particularly significant for the QAOA implementation, as it demonstrates that despite its computational overhead, the quantum approach consistently identified optimal tours.

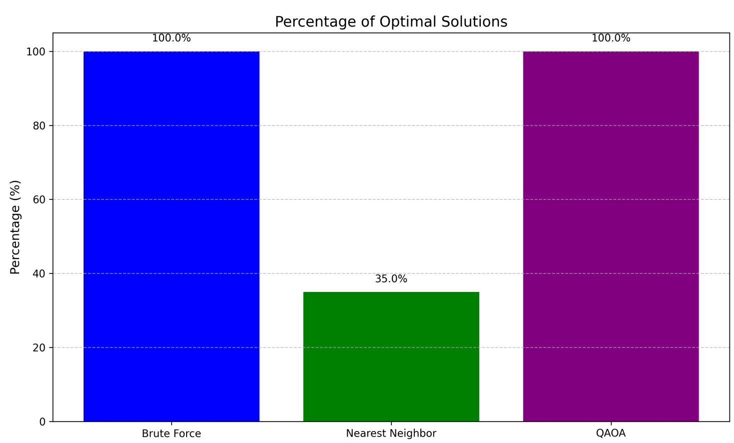

Figure 3: Percentage of trials resulting in optimal solutions

3.4 Time-Optimality Trade-off

Figure 4 illustrates the fundamental trade-off between execution time and solution optimality. This visualization reveals three distinct performance profiles: (1) Nearest Neighbor offers exceptional speed with moderate optimality, (2) brute force provides guaranteed optimality with reasonable speed for small instances, and (3) QAOA achieves perfect optimality but at significantly higher computational cost.

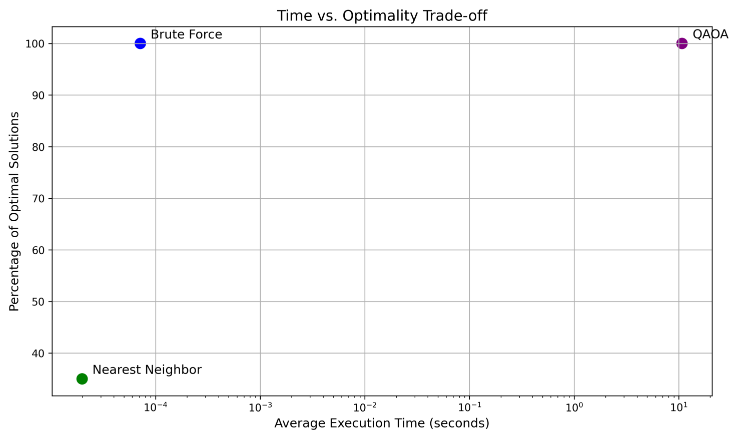

Figure 4: Time vs. optimality trade-off for all algorithms

3.5 Summary of Results

Table 1 summarizes the performance metrics across all three algorithms. The key finding is that our QAOA implementation successfully matched the solution quality of brute force in all instances, demonstrating the potential of quantum approaches for finding optimal solutions to combinatorial optimization problems. However, the current computational overhead remains a significant barrier to practical application.

Algorithm	**Avg. Time (s)**	**Avg. Cost**	**Optimal Solutions (%)**	**Cost Ratio**
Brute Force	0.000072	2.1219	100.0%	1.0000x
Nearest Neighbor	0.000020	2.2025	35.0%	1.0380x
QAOA	10.761290	2.1219	100.0%	1.0000x

Table 1: Summary of performance metrics across all algorithms

4 Discussion

4.1 Interpretation of Key Findings

Our experiments demonstrate that QAOA consistently achieved optimal solutions across all test instances, matching the performance of brute force in solution quality while requiring significantly more computation time (10.76 seconds vs. microseconds for classical approaches). This confirms that our quantum approach effectively navigates the solution space to find global optima despite its probabilistic nature.

The substantial execution time difference highlights the computational overhead of current quantum simulation, stemming primarily from the classical simulation of quantum circuits and the hybrid optimization process requiring multiple circuit evaluations. Meanwhile, Nearest Neighbor's profile—near-optimal solutions with minimal execution time—represents the pragmatic approach for instances where brute force becomes intractable.

4.2 Quantum Encoding and Scaling Challenges

A central challenge in our quantum implementation was efficiently encoding TSP solutions into qubit states. Our binary encoding approach required $n \times \lceil \log_2(n) \rceil$ qubits to represent city positions, resulting in 15 qubits for our 5-city problems. This scales more efficiently than one-hot encoding (n^2 qubits) but still faces exponential circuit complexity as problem size increases.

The relationship between problem size and quantum resources presents both opportunities and challenges:

Cities	**Qubits Required**	**Classical Solutions**	**Potential Advantage**
5	15	24	None
10	40	362,880	Limited
15	60	1.3×10^{11}	Moderate
20	100	2.4×10^{17}	Significant
25	125	6.2×10^{23}	Strong

Table 2: Scaling relationship between problem size and quantum resource requirements

As Table 2 illustrates, potential quantum advantage emerges at larger problem sizes where classical brute force becomes intractable. However, current NISQ (Noisy Intermediate-Scale Quantum) devices face significant limitations:

- Limited qubit counts (typically 50-100), constraining maximum problem sizes
- Short coherence times and gate errors affecting circuit performance
- Connectivity constraints complicating our entanglement patterns

While our 5-city experiments enabled direct comparison with brute force, the theoretical scaling advantages become relevant at larger problem sizes. We project that brute force becomes intractable beyond 12-15 cities, while QAOA's theoretical advantage could emerge at 20+ cities if hardware can support the required resources.

4.3 Future Outlook

Based on our findings, several promising research directions emerge:

- **Improved Encodings:** Developing more efficient qubit encoding strategies
- **Circuit Depth Reduction:** Creating shallower circuits with fewer parameters
- **Hardware-Specific Optimizations:** Tailoring implementations to specific quantum architectures
- **Hybrid Approach Enhancement:** Optimizing the classical component to reduce circuit evaluations

Our study demonstrates that while quantum approaches can achieve optimal TSP solutions, bridging the efficiency gap remains the key challenge for practical quantum advantage. The future of quantum TSP optimization likely resides at the intersection of hardware improvements, efficient encodings, and enhanced hybrid algorithms that leverage both quantum and classical strengths.