Exploring Global Identities Using Quantum Computing and GIS Project Report

Independent Study in Quantum Computing (CSCI 298)
Dr. Michael P. Haydock
Sat Barseghyan

Introduction

This project uses quantum computing to solve the Traveling Salesman Problem (TSP), which finds the shortest route between cities, with applications in logistics and transportation.

How did this project come to be? Well, I love traveling, I have personal connections to some of these cities, and honestly... I just think maps are really cool. Plus, I recently discovered that iOS Photos app map feature, and it blew my mind. I do think this project has the potential for bigger and better things, especially for real-world optimization, but, you know, time wasn't on my side. So here we are, with something cool, but slightly less world-changing... for now! Inspired by my experience living in multiple countries, I mapped cities significant to me to explore how global movement shapes identities. This combines quantum optimization with storytelling about cultural connections. Using the Quantum Approximate Optimization Algorithm (QAOA), the project demonstrates how quantum principles like superposition and entanglement solve complex problems. By integrating clustering and data visualization, it presents relationships between cities in an interactive and meaningful way.

2. Research and Background

This project explores how quantum computing can solve complex optimization problems by leveraging principles like superposition and entanglement to evaluate multiple solutions simultaneously. The focus on cities meaningful to my journey adds a personal layer to the project, connecting quantum tools with real-world storytelling.

Key Resources:

- Quantum Basics: Studies of gates, circuits, and algorithms like QAOA built the foundation.

Quantum Features Used

- A 10-qubit quantum device was simulated using PennyLane. Each qubit represented a city in the Traveling Salesman Problem (TSP).

Quantum Approximate Optimization Algorithm (QAOA)

- The optimization problem (TSP) was encoded into a quantum framework using Hamiltonians:
    - Cost Hamiltonian: Represented the total travel distance, which was minimized to solve the TSP.
    - Mixer Hamiltonian: Allowed the exploration of multiple solutions through quantum state mixing.
- **Implemented using:**
    - Rotation Gates (RX): Initialized qubit states and applied mixing based on the mixer Hamiltonian.
    - Entanglement Gates (CNOT): Encoded relationships between cities, representing dependencies in the cost Hamiltonian.

- Iteratively optimized parameters to minimize the cost function using a gradient descent approach.

$$\text{Cost} = \sum_{i=1}^{n} \text{Distance between cities}$$

Superposition and Entanglement

- Superposition: Explored multiple possible routes simultaneously.
- Entanglement: Captured dependencies between cities, enabling a more connected optimization landscape.

Quantum Measurement

- Optimized routes were extracted from the quantum circuit using Pauli-Z measurements.

Hybrid Approach

- Combined quantum optimization results with classical methods:
  - K-Means Clustering: Grouped cities based on their attributes, such as landmarks, population size, and universities.
  - Visualization Tools: Integrated results into classical data visualization platforms like Folium and ArcGIS to present routes and patterns interactively.

3. Initial Code

The foundation of this project was built upon Dr. Michael P. Haydock's example code, which served as both a technical and conceptual guide. Key contributions from the code include:

- **Quantum Circuit Implementation**:
  - The professor's code provided a working framework for setting up quantum circuits, which were adapted and extended for this project.
  - It introduced the use of **rotation gates (RX)** and **entanglement gates (CNOT)** for encoding relationships between cities.
- **Optimization Techniques**:
  - The original implementation used the **COBYLA** optimization method, which inspired the adoption of gradient-based optimization techniques for tuning parameters in the Quantum Approximate Optimization Algorithm (QAOA).
- **Scalability**:
  - While the example code focused on a small number of cities with a fixed distance matrix, it motivated the shift to real-world data, incorporating coordinates, attributes, and dynamic clustering

4. Implementation and Results - Data Preparation

The first step of the project involved compiling a structured dataset of cities, their coordinates, and key attributes. The selected cities represented locations significant to global movement and personal identity, such as Yerevan, Maastricht, Northfield, and Newcastle upon Tyne.

- **Geographic Coordinates**: Latitude and longitude for accurate mapping.
- **Attributes**: Each city was characterized by features such as the presence of landmarks, population size, and the number of universities.

Quantum Optimization Workflow

1. **Quantum Circuit Design**:
   - Each city was represented by a qubit in a 10-qubit quantum device.
   - Rotation gates (RX) initialized the qubits to explore potential solutions.
   - CNOT gates created entanglement between qubits, encoding relationships between cities.
2. **Cost Function**:
   - A cost function was defined to minimize the total travel distance between the cities.
3. **Optimization**:
   - Parameters were iteratively adjusted using a gradient descent optimizer until the cost function converged to a minimum.
   - The result was an optimized travel route connecting all 10 cities with the shortest possible path.

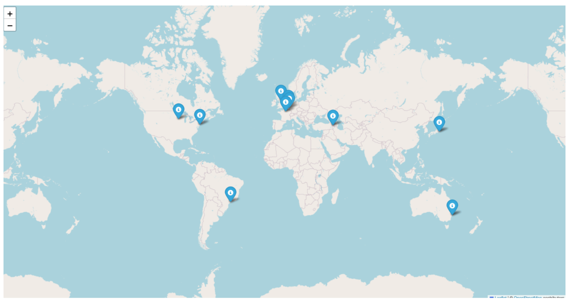

Figure 1. Base Map

Clustering Analysis

To further analyze the cities, their attributes were clustered using **K-Means clustering**. This revealed patterns such as:

- Cities with shared cultural diversity or large populations formed distinct groups.
- Cities with fewer attributes, such as landmarks or universities, were grouped together.

Visualization with ArcGIS

Visualization was a critical component of the project, enhancing its clarity and impact. Some of my key steps included:

1. **Exporting Data**:

   - The optimized route and city attributes were exported as a CSV file, containing details such as latitude, longitude, route order, and attributes for each city.
2. **Creating Interactive Maps**:

   - The data was uploaded to **ArcGIS Online**, where it was transformed into interactive maps.
   - The maps featured:
     - **Markers**: Representing each city with popups to display attributes.
     - **Optimized Routes**: Lines connecting cities in the order determined by the quantum algorithm.
     - **Dynamic Layers**: Allowing users to toggle between attributes, such as landmarks or population size.

3. **Impact**:

    ○ The ArcGIS maps provided a visually compelling way to present the results, making the project accessible to a wider audience.
    ○ This exploration also introduced new skills in GIS, showcasing the potential for integrating geographic visualization with quantum solutions.

Result:

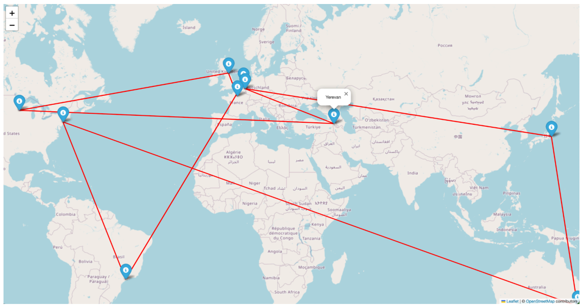

Figure 2. Optimized Map

1. **Optimized Route**: The quantum optimization algorithm successfully computed an efficient travel route between the cities, minimizing total distance.
2. **Clusters**: K-Means clustering highlighted meaningful groupings based on city attributes, adding depth to the analysis.
3. **Maps**: Interactive ArcGIS maps brought the project to life, combining optimization results with rich geographic and cultural insights.

5. ArcGIS

This project marked my first experience with **ArcGIS**, a powerful tool for geographic data visualization. It played a critical role in transforming the quantum optimization results into interactive, visually engaging maps. Key learnings included:

- **Uploading Datasets**: Successfully uploading the optimized travel routes and city attributes as a CSV file to create hosted feature layers.
- **Dynamic Visualization**: Creating maps with toggles for different city attributes, such as landmarks, population size, and universities.
- **Interactive Features**: Adding route connections and clustering information to make the maps more intuitive and engaging.
- **Challenges**: Navigating ArcGIS's interface and dynamically connecting route lines required troubleshooting, but the process provided valuable insights into GIS workflows.

6. Conclusion and Findings

This project successfully demonstrated the potential of **quantum optimization** for solving the Traveling Salesman Problem (TSP) using real-world data. By incorporating cities and their attributes, it provided valuable insights into the relationships between geographic locations and identity. Additionally, the integration of clustering analysis and visualization tools like **ArcGIS** showcased how classical tools can complement quantum computing to produce practical, meaningful results.

Key Findings:

- **Quantum Optimization**: Effectively minimized travel costs, highlighting the promise of quantum algorithms for complex optimization problems. However, challenges in scaling remain, particularly as the number of cities and attributes increases.
- **Clustering Analysis**: Revealed significant patterns and relationships between cities, demonstrating how quantum-inspired solutions can offer deeper insights into real-world data.
- **Visualization**: Tools like ArcGIS proved invaluable for interpreting results, making them accessible and engaging for a broader audience.

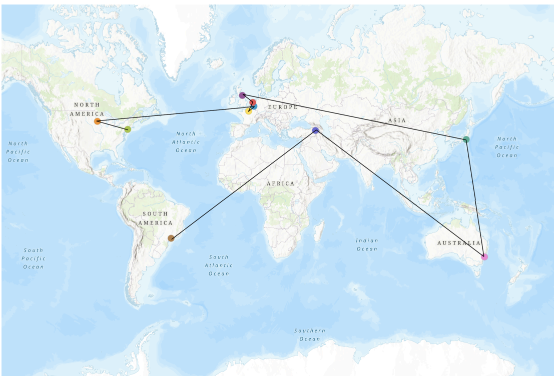

Figure 3. Final Map, displayed through ArcGIS

7. Future Explorations

1. Accounting for Real-World Costs

A significant extension would involve incorporating **real-world travel costs**, such as airfare, accommodation, and transportation. By including these economic factors, the project could produce more actionable insights for industries like travel agencies or airlines, enabling them to optimize their services and predict customer preferences.

2. Expanding the Dataset

Scaling the project to include more cities and attributes would enhance its depth. Future iterations could:

- Add cities from diverse global regions to reflect broader patterns of movement.
- Include additional attributes such as economic activity, cultural events, or climate preferences to improve clustering and analysis.

3. Advanced Quantum Algorithms

The project currently focuses on the Quantum Approximate Optimization Algorithm (QAOA) for route optimization. Future work could explore other algorithms, such as:

- **Grover's Algorithm**: For efficient searching within large datasets.
- **Variational Quantum Classifiers (VQC)**: To enhance clustering and pattern recognition in city attributes.

Exploring Variational Quantum Classifiers (VQC)

While not implemented in this project due to time constraints, code for a Variational Quantum Classifier (VQC) was developed to explore its potential for clustering cities based on their attributes. A VQC uses parameterized quantum circuits to classify data and could offer more nuanced insights compared to classical clustering methods like K-Means.

```python
# Data Encoding

def encode_features(features):

    """Normalize and encode city attributes for quantum input."""

    norm_features = np.array(features) / np.max(features, axis=0)  # Normalize

    return norm_features * np.pi  # Scale to [0, pi]

# Define Quantum Circuit

@qml.qnode(qml.device("default.qubit", wires=len(features)))

def variational_circuit(params, features):

    """Variational Quantum Classifier circuit."""

    for i in range(len(features)):

        qml.RX(features[i], wires=i)  # Encode data

    for param in params:

        for i in range(len(features)):

            qml.RY(param[i], wires=i)  # Parameterized gates

        for i in range(len(features) - 1):

            qml.CNOT(wires=[i, i + 1])  # Entanglement

    return [qml.expval(qml.PauliZ(i)) for i in range(len(features))]

# Train the VQC

def loss(params, features, labels):

    """Loss function for VQC training."""

    predictions = variational_circuit(params, features)

    return np.mean((predictions - labels) ** 2)  # Mean squared error

# Example training workflow

init_params = np.random.rand(len(features), len(features))
```

```python
opt = qml.GradientDescentOptimizer(stepsize=0.1)

for step in range(100):

    init_params = opt.step(lambda p: loss(p, encoded_features, labels), init_params)

# Predict Clusters

predictions = variational_circuit(init_params, encoded_features)

clusters = np.round(predictions)  # Clusters based on predictions
```

Why VQC?

- **Quantum Advantages**: Unlike classical clustering methods, a VQC leverages quantum superposition and entanglement to identify subtle correlations.
- **Scalability**: As quantum hardware matures, VQC methods could outperform classical algorithms for larger datasets.
- **Practical Use**: These clusters could be visualized alongside travel routes to enrich the project's results and applications.

5. Enhanced Visualization

- Adding animations or dynamic dashboards to showcase routes interactively.
- Enabling toggles for real-world metrics like travel costs or environmental impact.

8. Bibliography

1. Michael P. Haydock. "Quantum TSP Code Example." Independent Study in Quantum Computing, Fall 2024, St. Olaf College.

2. PennyLane Documentation. *PennyLane: Machine Learning with Quantum Circuits*. Available at: https://pennylane.ai/.
3. ArcGIS Online Documentation. *Introduction to ArcGIS Online*. Available at: https://www.arcgis.com/home/index.html.
4. D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization." *arXiv preprint arXiv:1412.6980*, 2014. Available at: https://arxiv.org/abs/1412.6980.
5. Krishnakumar, Type. "What is a Variational Quantum Classifier?" *Medium*, 2023. Available at: https://medium.com/@typekrish/what-is-a-variational-quantum-classifier-888e40f83b24.
6. GeeksforGeeks. "Traveling Salesman Problem Using Dynamic Programming." *GeeksforGeeks*, 2023. Available at: https://www.geeksforgeeks.org/travelling-salesman-problem-using-dynamic-programming/.
7. Qiskit. "Building a Quantum Variational Classifier Using Real-World Data." *Medium*, 2023. Available at: https://medium.com/qiskit/building-a-quantum-variational-classifier-using-real-world-data-809c59eb17c2.
8. Recent Advances in Quantum Optimization Algorithms. *arXiv preprint arXiv:2407.17207*, 2024. Available at: https://arxiv.org/abs/2407.17207.