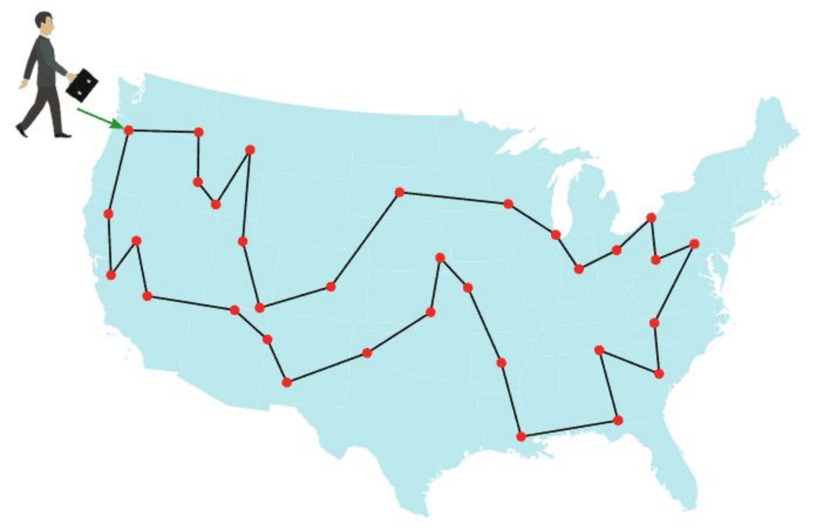
# Travelling Salesman Problem via Quantum Annealing

Jaswanth Duddu
Matthew Tavares
Zachary Bonagura

# What is TSP?

Objective: find the shortest possible path allowing a salesman to visit given cities exactly once before returning to original city.



NP-Hard problem

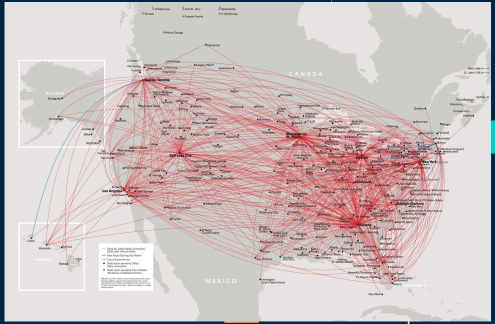
Classical algorithms struggle to find optimal solutions efficiently, opening the potential of a quantum approach.

# Applications of TSP

- Delivery Routing
- Warehouse Picking
- Machine Scheduling
- Airline Scheduling
- Railroad Scheduling
- Network Design
- Genome Sequencing
- Public Transit
- Crop Spraying

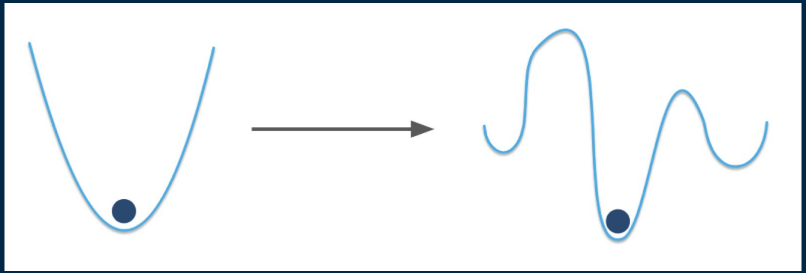




# Quantum Annealing

- Annealing: optimize solutions to problems by quickly searching a space, and finding the global minimum, which is the solution.
- Quantum annealing focuses on solving optimization problems, and generally not based on quantum circuits.
- We will be using adiabatic quantum computing:
    - Start with simple optimized solution.
    - Space is explored and changes to a more complicated one.
    - We can still guarantee that it will remain the minimum.

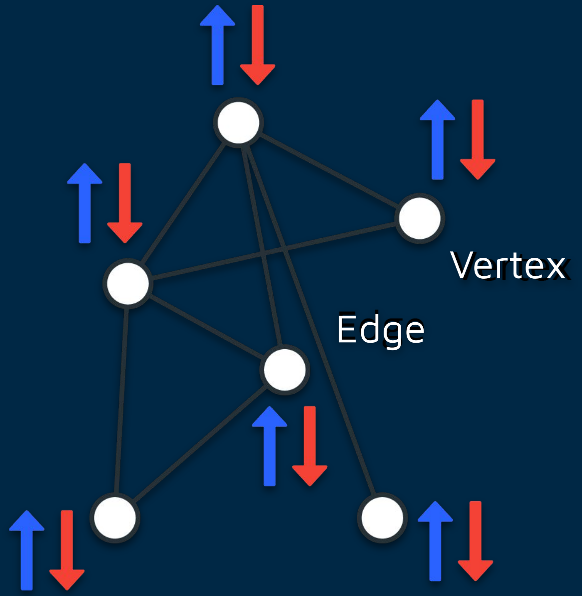
# Hamiltonian

| Energy | | Initial Hamiltonian | | Final Hamiltonian |
|---|---|---|---|---|
| $\mathcal{H}_S(s)$ | $=$ | $-\frac{1}{2}\sum_i \Delta(s)\sigma_i^x$ | $+$ | $\mathcal{E}(s)\left(-\sum_i h_i\sigma_i^z + \sum_{i<j} J_{ij}\,\sigma_i^z\sigma_j^z\right)$ |

- We can model quantum annealing with the Hamiltonian, known as the ground-state problem.
  - Hamiltonian gives the total energy of a system.
  - We want to find the state of qubits that give lowest energy of a system.
- **Energy**: What we're calculating.
- **Initial**: Lowest energy state is where qubit is in superposition of 0 and 1.
- **Final**: As you anneal, problem Hamiltonian is introduced.
  - Eventually, as it reaches the end of the anneal, the Energy becomes solely the final.
- You've stayed in minimum energy state the entire time!

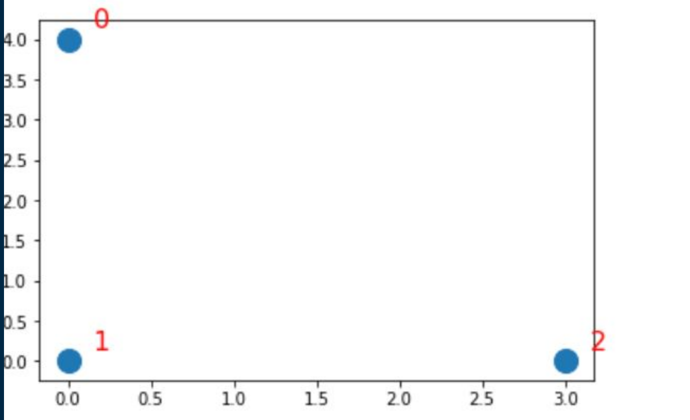
# Plan

1. Mapping TSP to Ising Model
2. Formulate Ising Model into QUBO
3. Apply Quantum Annealing
4. Validate Results



# Demonstration

```python
from scripts import utilities
from scripts import plots
import matplotlib.pyplot as plt
# cities = utilities.create_cities(3)
cities = np.array([[0, 4],[0, 0],[3, 0]])
distance_matrix = utilities.get_distance_matrix(cities)
plots.plot_cities(cities)
plt.show()
```




```python
distance_matrix = utilities.get_distance_matrix(cities)
print(distance_matrix)
number_of_cities = len(cities)
```

```
[[0. 4. 5.]
 [4. 0. 3.]
 [5. 3. 0.]]
```

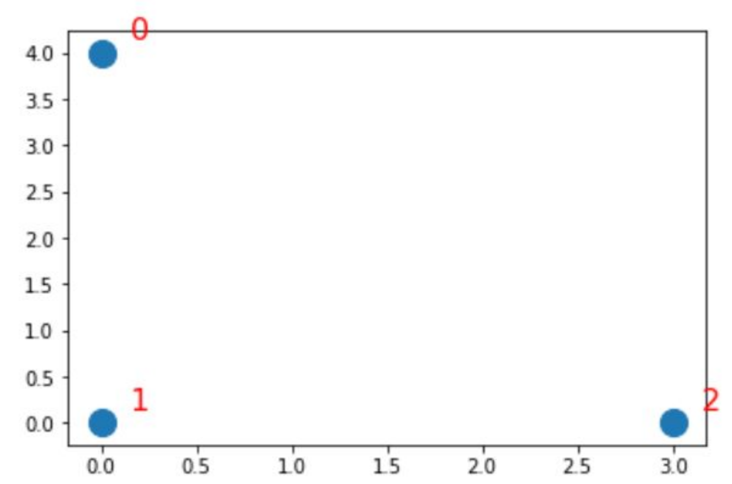
The last time we learned two useful equations for MaxCut which will guide us now.

One for the classical cost function:

$$C_{total} = \sum C_{ij} = \sum \frac{1}{2} w_{ij}(1 - z_i z_j),$$

And one showing how to construct the quantum cost operator:

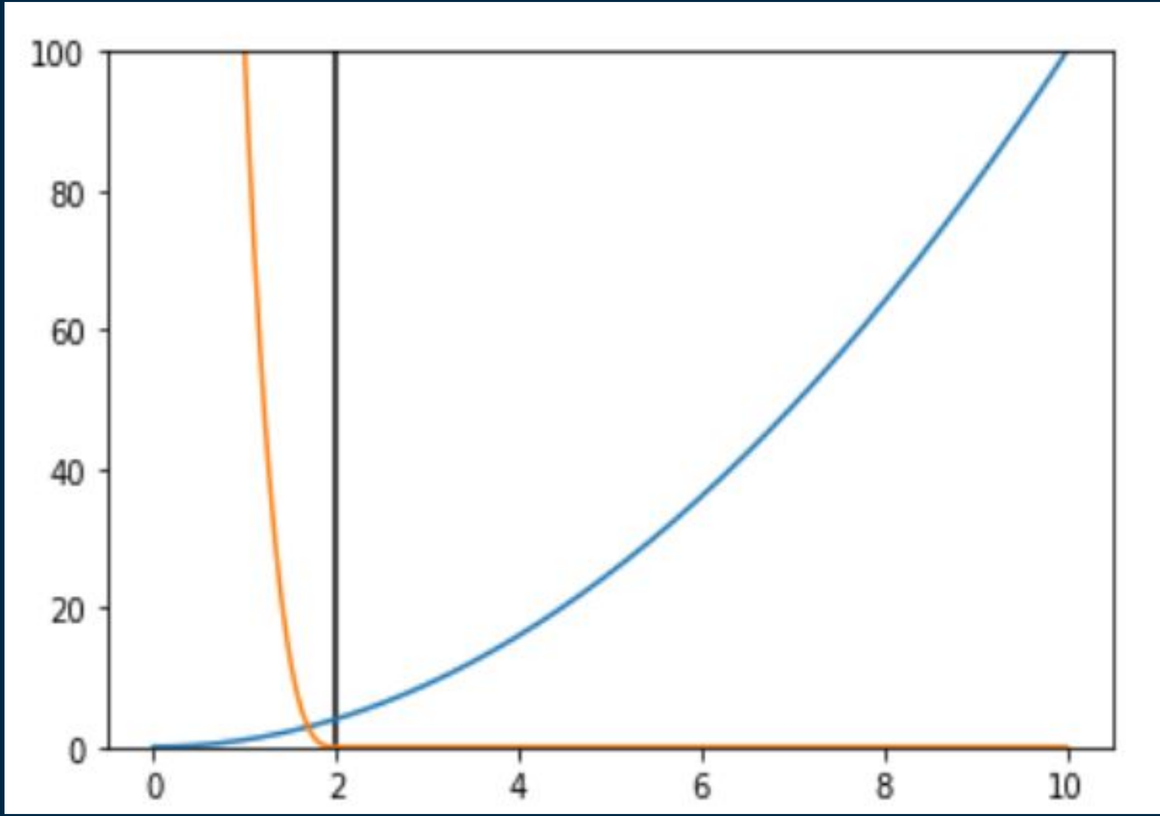
$$H_{cost} = \sum \frac{1}{2} w_{ij}(\mathbb{1} - \sigma_i^z \sigma_j^z)$$




```
City 0 to 1 at t = 0 costs -2.0 Qubits: 0 4
City 0 to 1 at t = 1 costs -2.0 Qubits: 3 7
City 0 to 2 at t = 0 costs -2.5 Qubits: 0 5
City 0 to 2 at t = 1 costs -2.5 Qubits: 3 8
City 1 to 2 at t = 0 costs -1.5 Qubits: 1 5
City 1 to 2 at t = 1 costs -1.5 Qubits: 4 8
```

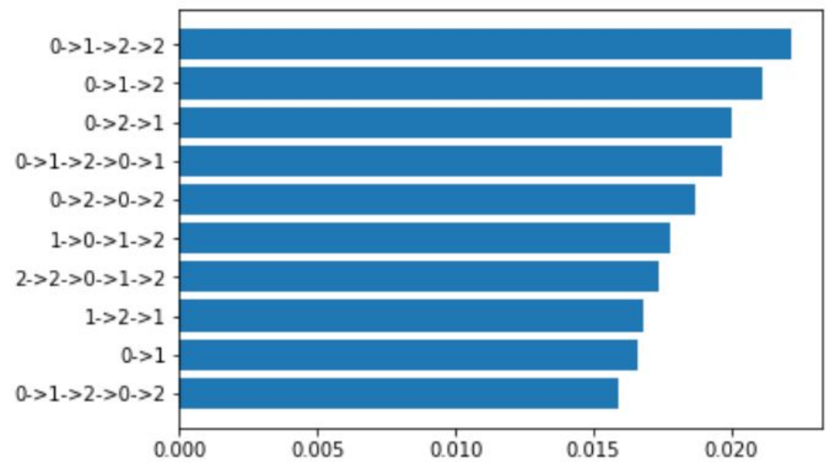
```python
for single_cost_operator in cost_operators:
    print(single_cost_operator)
```

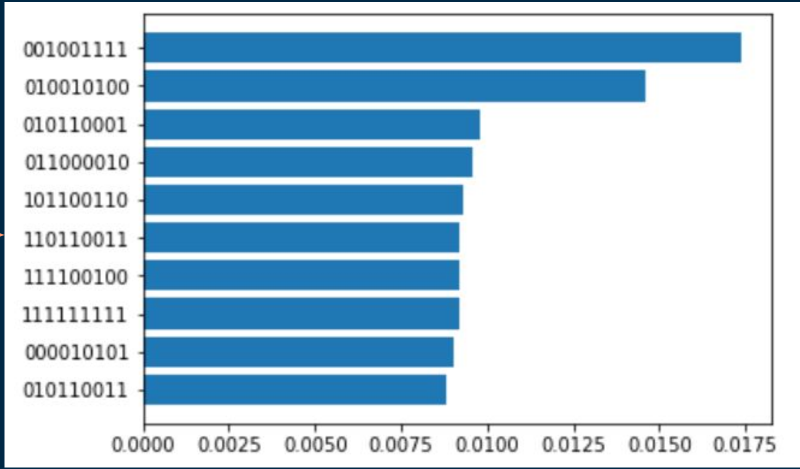
```
(-2+0j)*I + (2-0j)*Z0*Z4
(-2+0j)*I + (2-0j)*Z3*Z7
(-2.5+0j)*I + (2.5-0j)*Z0*Z5
(-2.5+0j)*I + (2.5-0j)*Z3*Z8
(-1.5+0j)*I + (1.5-0j)*Z1*Z5
(-1.5+0j)*I + (1.5-0j)*Z4*Z8
```


100
80
60
40
20
0
0
2
4
6
8
10

Classical TSP

Quantum TSP



# References


- https://arxiv.org/pdf/1805.10928
- https://arxiv.org/pdf/2405.07129
- https://arxiv.org/pdf/2307.07243
- https://arxiv.org/pdf/2407.07202
- https://helda.helsinki.fi/server/api/core/bitstreams/d993a874-20c1-4956-a203-2c2e909bd0d3/content
- https://learn.microsoft.com/en-us/azure/quantum/concepts-grovers
- https://journals.aps.org/pre/pdf/10.1103/PhysRevE.70.057701
- https://www.baeldung.com/cs/tsp-exact-solutions-vs-heuristic-vs-approximation-algorithms
- https://qiskit-community.github.io/qiskit-optimization/tutorials/06_examples_max_cut_and_tsp.html#
- https://github.com/mstechly/quantum_tsp_tutorials/tree/master