Solving the Travelling Salesman Problem on IBM Quantum®

Natália Capra Ferrazzo

Institute of Physics, Federal University of Rio Grande do Sul, Porto Alegre, Brazil, natalia.ferrazzo@ufrgs.br.

Abstract. In this article, the basic concepts of Quantum Computing were studied and applied to NP-hard Travelling Salesman problem. Through the tools provided by IBM Quantum and Qiskit initiative, it was able to implement it via Python libraries and Jupyter Notebook. The results were as computed for similar publications.

Keywords: NP-hard, Quantum computing, Travelling Salesman Problem

1. Introduction

As Richard Feynman highlighted, some quantum mechanics effects, as superposition and entanglement, can not be simulated efficiently on classical computers [1], raising the conjecture that computation can be done more efficiently if quantum effects are used to perform it. Specifically, as mentioned by Acompora et al. [2], Feynman speculated that the synergistic usage of quantum superposition and entanglement in computation may enable the design of computing devices showing a high degree of parallelism, which grows with the size of the device itself, in an exponential way.

In this way, quantum computers are being used to approach these kinds of problems using various techniques, even though there are supercomputers all over the world. It's because of the fact of what a quantum computer can do over a classical computer, using the weirdness of quantum mechanics and all the different properties which only a quantum computer can provide, and using algorithms developed for quantum computer over the years.

Although the use of quantum computers doesn't guarantee the problem will be solved, it still offers a new way to approach problems of this class.

2. Basic Concepts

2.1 Qubits

Classical computing is based on *bit*: a binary digit that can be in one of two mutual exclusive states, either 0 or 1. Quantum computing, otherwise, is based on *qubit*: it can be in a quantum state that is a complex linear superposition of 0 and 1, before being measured. When it is measured, it *"collapses"* to one of these two values, corresponding to classical bits. When a qubit is in a super-position of states, it can be said that it has an *amplitude* associated with each state.

Formally, a qubit is represented by a unit vector, $|\psi\rangle$ of a two-dimensional Hilbert space:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

Where $\alpha, \beta \in \mathbb{C}$ and it's normalized so:

$$|\alpha|^2 + |\beta|^2 = 1$$

In this way, $|\alpha|^2$ can be interpreted as the *probability of measuring* $|0\rangle$ and $|\beta|^2$ as the *probability of measuring* $|1\rangle$.

The $|0\rangle$ and $|1\rangle$ are the basis states of the Hilbert space and can be represented by:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

2.2 Bloch Sphere

There is an alternative representation that allows a better visualization of a qubit – the so-called *Bloch Sphere*. In order to design it, it's needed to use the following representation of a qubit, derived from the polar form of complex numbers:

$$cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi} sin\left(\frac{\theta}{2}\right)|1\rangle$$

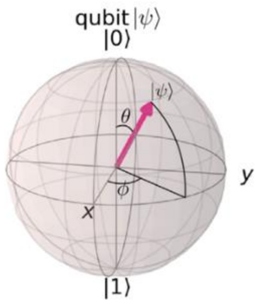


Fig 1. Bloch sphere

2.3 Quantum gates

The evolution of a closed quantum system is described by special linear operators, *unitary operators* U which operate on qubits as follows:

$$U|\psi\rangle = \mathrm{U}[\alpha|0\rangle + \beta|1\rangle] = \alpha\mathrm{U}|0\rangle + \beta\mathrm{U}|1\rangle$$

The most common gate is the NOT-gate, or, in quantum equivalence Pauli-X gate. It has the form of:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Suppose a qubit in state $|\psi\rangle = 1 \cdot |0\rangle + 0 \cdot |1\rangle$. If one compute $|\psi'\rangle = X|\psi\rangle$, gets:

$$|\psi'\rangle = 1 \cdot \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes 0 \cdot \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$|\psi'\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

The most interesting outcome of Pauli-X gate occurs when it is applied to a qubit in superposition state. In such case, it inverts the probability that the quantum state will collapse to 0 or 1.

To create a superposition state, a Hadarmard gate can be applied, the unitary operator is:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Suppose, again, a qubit in state $|\psi\rangle = 1 \cdot |0\rangle + 0 \cdot |1\rangle$. If one compute $|\psi'\rangle = H|\psi\rangle$, gets:

$$|\psi'\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \left(1 \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0 \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right)$$

$$|\psi'\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$|\psi'\rangle = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$$

As one can see, after applying the quantum operator H, the qubit will be in a superposition state:

$$|\psi\rangle = \frac{1}{\sqrt{2}} \cdot |0\rangle + \frac{1}{\sqrt{2}} \cdot |1\rangle$$

In this case, the probability that after measuring the qubit, it will be found in state $|0\rangle$ or $|1\rangle$ is the same: 50%.

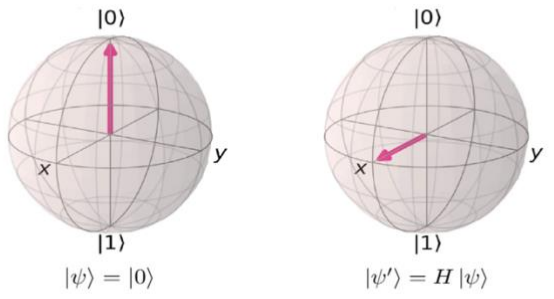


Fig 2. Hadamard gate applied to a qubit in state $|0\rangle$: $\mathrm{H}|0\rangle$

Other interesting quantum gates are R_x, R_y and R_z, because they allow a simple and direct modification of the magnitude and phase.

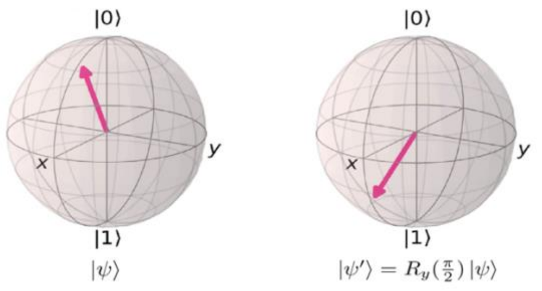


Fig 3. R_y gate applied to a qubit in state $|\psi\rangle$.

For example, R_y performs a single-qubit rotation through angle θ radians around the y-axis. The unitary operator associated with the R_y gate is:

$$R_y(\theta) = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}$$

The R_y rotation mainly acts on the magnitude knob of qubit. Thus, this gate modifies the probability that a qubit in state $|\psi\rangle$ will collapse to 1 or 0, after measuring it. **Fig 3** shows an example of the application of the gate $R_y(\frac{\pi}{2})$ to a generic quantum state $|\psi\rangle$.

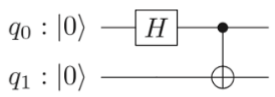
A very special phenomenon takes place when a quantum device has access to more than one qubit: *entanglement*.

One can get an entangled state by applying a Hadamard gate followed by a *CNOT* gate, which operator is given by:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The CNOT gate operates on two qubits, a *control* qubit and a *target* qubit, and it works as it: apply the logical NOT operation to the target qubit only if the control qubit has the value 1.

So, if a Hadamard gate is applied in q_0 and then a CNOT whit a control qubit as q_0 and target as q_1, an entanglement state is created.



The result is a superposition of $|00\rangle$ and $|11\rangle$, which cannot be represented as two separated qubits anymore: they are *entangled*. In other words, the value of q_1 is completely connected to the quantum measurement on q_0.

3. IBM Quantum®

IBM provides some useful tools for quantum computing. In this report, IBM Quantum lab was used for implementing all the algorithms, and an introduction for each tool is provided below.

3.1 IBM Quantum Composer®

One can easily graphically build quantum circuits with IBM Quantum Composer. It allows users to dynamically see some useful circuit properties like the outcome probabilities of measurement, statevectors, phase and Q-sphere. In addition, it provides the Qiskit/Python implementation for the currently circuit.

3.2 IBM Quantum Lab®

This toolkit provides a collection of Jupyter Notebooks tutorials created by Qiskit® team. Also, one can creates its own notebook directly in this pane.

4. Travelling Salesman Problem

The Travelling Salesman Problem (TSP) belongs to the class of NP-Hard problems in combinatorial optimization. The problem is:

"Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?"

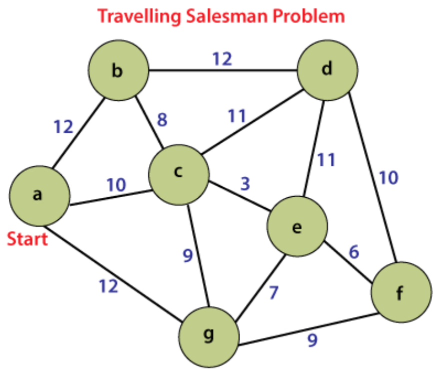


Fig 4. TSP graph representation: The letters represent cities, and the lines values are the distances between them.

The Travelling salesman problem is very important in theoretical computer science. Community has been trying to find an algorithm which can solve it in polynomial time. The reason of its importance is not specifically TSP, but instead the class to which TSP belongs to. It's *the NP class*. If one can find an algorithm for TSP, it will open a wide range of possibilities for the thousands of other problems which belongs to the same class.

The TSP problem is similar to the Hamiltonian cycle problem, as follows:

"The cycle of visiting each vertex once in a graph and returning to the starting vertex is known as a Hamiltonian cycle. Given a graph, determine whether it contains a Hamiltonian cycle or not."

As the Hamiltonian cycle is at the heart of the TSP, these two problems are interconnected and solving one will solve the other.

5. Methodology

To solve TSP, one can consider the whole problem in terms of graphs. The cities are represented as vertices, and the cost/path as edges.

- The problem is approached by encoding the given distances/cost between the cities as phases.
- Each city is connected to other cities with a specific cost associated to each connection.
- Unitary operators are constructed whose eigenvectors are the computational basis states and eigenvalues are various combinations of these phases.
- Then its applied phase estimation algorithm to certain eigenstates which gives all the total distances possible for all the routes.
- After obtaining the distances, a search is made through this information using the quantum search algorithm to find the least possible distance as well the route taken.

This approach provides a quadratic speedup over the classical brute force method for a large number of cities.

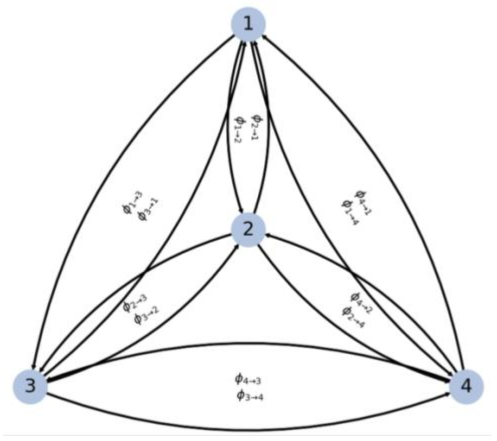


Fig 5. Hamiltonian cycle representation with 4 nodes.

5.1 Phase Estimation

The *quantum phase estimation algorithm* uses *phase kickback* to write the phase of U to the t qubits. When a qubit is used to control the U gate, the qubit will turn proportionally to the phase $e^{2i\pi\theta}$. It's needed to introduce the controlled unitary – CU – that applies the unitary operator U on the target register only if its corresponding control qubit is $|1\rangle$. So, it's needed to decompose these unitaries as *controlled unitaries*. As follows:

$$U_1, U_2, U_3, U_4 \to CU_1, CU_2, CU_3, CU_4$$

$$U_j = \begin{bmatrix} e^{ia} & 0 & 0 & 0 \\ 0 & e^{ib} & 0 & 0 \\ 0 & 0 & e^{ic} & 0 \\ 0 & 0 & 0 & e^{id} \end{bmatrix}$$

The eigenvalues of this unitary matrix U are estimated using the quantum phase estimation algorithm.

The phases can be normalized to be bound within 0 and 2π once the range of distances between the cities is known.

U is a diagonal matrix since it is a tensor product of n diagonal matrices. This means that the eigenstates of this matrix U are computational basis states with eigenvalues as the corresponding diagonal elements.

The eigenstates are represented in binary form to convert the city to computational basis vectors using a function:

$$|\psi\rangle = \otimes_j |\mathrm{i(j)} - 1\rangle \quad \text{where j } \in [1 \cdots \mathrm{n}]$$

where the function i(j) corresponds to *"from which city the salesman travelled to city j"*?

5.2 Eigenstates

With 4 cities taken, the total combination of all possible Hamiltonian cycle is $\boldsymbol{n = 4! = 24}$. Out of these 24, 6 are *distinct* Hamiltonian cycle.

We are taking the first 6 states, since these states on different circular permutations will give all the states:

Possible States
$1-2-3-4$
$1-2-4-3$
$1-4-2-3$
$1-4-3-2$
$1-3-2-4$
$1-3-4-2$

Tab 1. Possible distinct states

In the binary form, they assume the form:

Sequence path	Eigenstates
$1-2-3-4$	$\|11000110\rangle$
$1-2-4-3$	$\|10000111\rangle$
$1-4-2-3$	$\|10001101\rangle$
$1-4-3-2$	$\|01001110\rangle$
$1-3-2-4$	$\|11001001\rangle$
$1-3-4-2$	$\|01001011\rangle$

Tab 2. Eigenstates in binary form

Considering the same cost from going and returning, there will be 3 sequence paths out of 6. For example, we can take any one of 1−2−3−4 or 1−2−4−3, since cost from 3−4 will be the same as 4−3. So, these 6 will reduce to:

Sequence path	Eigenstates
$1-2-3-4$	$\|11000110\rangle$
$1-4-2-3$	$\|10001101\rangle$
$1-3-2-4$	$\|11001001\rangle$

Tab 3. Eigenstates reduced

6. Results

The phases are normalized to be bound within [0, 2π] once the range of distances between the cities are known.

Phases	Encoded as	Phase value
$\phi_{1\to1}$		0
$\phi_{2\to1}$	a	$\frac{\pi}{2}$
$\phi_{3\to1}$	b	$\frac{\pi}{8}$
$\phi_{4\to1}$	c	$\frac{\pi}{4}$
--	--	--
$\phi_{1\to2}$	d	$\frac{\pi}{2}$
$\phi_{2\to2}$		0
$\phi_{3\to2}$	e	$\frac{\pi}{4}$
$\phi_{4\to2}$	f	$\frac{\pi}{4}$
--	--	--
$\phi_{1\to3}$	g	$\frac{\pi}{8}$
$\phi_{2\to3}$	h	$\frac{\pi}{4}$
$\phi_{3\to3}$		0
$\phi_{4\to3}$	i	$\frac{\pi}{8}$
--	--	--
$\phi_{1\to4}$	j	$\frac{\pi}{4}$
$\phi_{2\to4}$	k	$\frac{\pi}{4}$
$\phi_{3\to4}$	l	$\frac{\pi}{8}$
$\phi_{4\to4}$		0

Tab 4. Phases normalized

The circuit was built and drawn in a Jupyter Notebook.

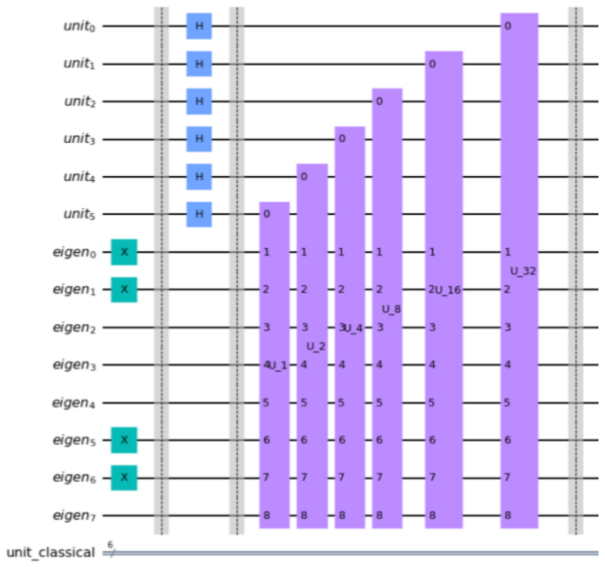


Fig 6. Circuit Implemented - *Part 1*

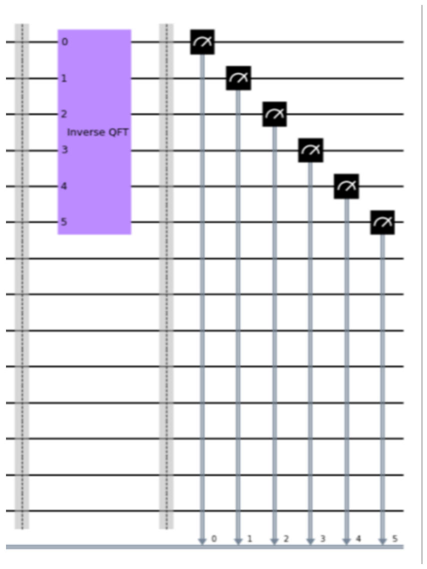


Fig 7. Circuit implemented - *Part 2*

And the result was computed. The number of shots taken was 8129 and in 100% of it, the result was the same: |100100⟩.

```
{'100100': 8192}
```

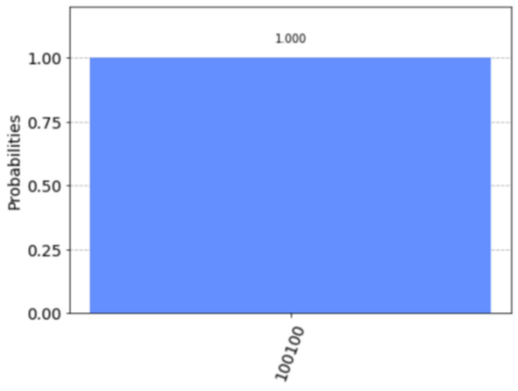


Fig 8. Final measurement.

7. Discussion

This process must be done for all the eigenstates to find the total distance for all the routes. After that, one can use quantum search algorithms to find the minimum of those distances. Thus, the time required scales with the number of eigenstates.

The paper token as reference for this has many circuit errors. They were fixed and the improved version was used.

A good question must be - Does using this process of quantum phase estimation to solve a problem of NP-Hard give a more efficient and optimal algorithm to solve these kinds of problems? No. This process of using phase estimation, which encode the cost as phases is one of the many ways to solve this category of problems. In this process, it was found all the possible Hamiltonian cycle in the graph and based on those cycle, the total cost was calculated. But finding all Hamiltonian cycle of a graph is itself a *NP-Complete* problem. It was used precomputed

Hamiltonian cycle in the beginning to get the eigenstates. And just 4 nodes/cities were used, but for large number of cities, finding all possible Hamiltonian cycle is an extensive work.

8. Conclusion

As mentioned by Srinivasan et al. [4], using the proposed algorithm, it was able to create a database of all possible routes that can be taken along with the distance of each. If one devises a quantum algorithm to find the minimum element in an unsorted array, which is faster than the one it's currently used, then one can use that algorithm to find the minimum. This gives this algorithm flexibility, which then can be exploited in the future to solve the travelling salesman problem much more efficiently. Even though this algorithm deals with a very general case, there are certain cases which cannot be directly solved using it. These are the cases where there are restrictions on routes connecting cities. For instance, city *b* does not have a route connecting it to city *d*. This can be thought of as the distance between those cities being infinite. Since this algorithm requires distances that can be normalized such that the total distance for the longest route is less than 2π, this does not bode well for it.

9. References

[1] R.P. Feynman, Simulating physics with computers, Int. J. Theor. Phys. 21 (6) (1982) 467–488.

[2] G. Acampora and A. Vitiello. Implementing evolutionary optimization on actual quantum processors, Int. N. Fisica Nucleare. (2021).

[3] P.W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput. 26 (5) (1997) 1484–1509.

[4] Karthik Srinivasan, Saipriya Satyajit, Bikash K Behera, and Prasanta K Panigrahi. Efficient quantum algorithm for solving travelling salesman problem: An IBM quantum experience. arXiv:1805.10928v1, 2018.

[5] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John Smolin, Harald Weinfurter. Elementary gates for quantum computation. arXiv:9503016v1, 1995.

[6] Qiskit Textbook. Solving the Travelling Salesman Problem using Phase Estimation. Avaliable on: https://qiskit.org/. Access date: Apr, 14th 2022.