A quantum approach to the travelling salesman problem.

Pau Torrente Badia

July 2023

1 Introduction

The travelling salesman problem, or TSP for short, is a very well known NP-hard problem that consists of finding the Hamiltonian cycle of a graph, this being a path that covers all nodes just once and returns to the starting point, that minimizes the overall distance travelled. While the formulation of the problem is quite simple, and finding the solution for simple graphs can be quite easy, for larger ones it is not so trivial to find the shortest Hamiltonian cycle (it scales non-polynomially). Since many real problems, especially related to optimization and logistics, can be mapped into this kind of mathematical problem, it can be really useful to be able to find solutions to the TSP to improve on efficiency and effectiveness in real world scenarios.

In this work we will try to tackle a few examples of the problem using quantum methods by first encoding the constraints and objectives of the problem as a Hamiltonian, and then finding its ground state, which will encode a solution. To do so we will work with 2 different quantum algorithms/procedures, one analog and one digital, to compare the two paradigms of quantum computing.

2 Problem encoding

The first, and most important step to solve the traveling salesman problem in our context is to encode it correctly in a mathematical way by constructing a Hamiltonian. For a graph with N nodes we will have N^2 degrees of freedom to determine, regarding the N different positions in a Hamiltonian cycle that each node can occupy. We will therefore describe our system with the variables $x_{u,j}$, referring to if node u occupies position j in a given cycle or not.

If we recall the description of the TSP, its different constraints and the cost function to be minimized can be encapsulated as the following Hamiltonian:

$$H_{tsp} = \sum_{u=1}^{N}\left(1-\sum_{j=1}^{N}X_{u,j}\right)^2 + \sum_{j=1}^{N}\left(1-\sum_{u=1}^{N}X_{u,j}\right)^2 + \sum_{(u,v)\notin E}\sum_{j=1}^{N}X_{u,j}X_{v,j+1} + \sum_{(u,v)\in E}W_{uv}\sum_{j=1}^{N}X_{u,j}X_{v,j+1} \tag{1}$$

Where $X_{u,j}$ is the operator that when acting on a state describing the system has $x_{u,j} = 0, 1$ as its eigenvalues, regarding to if node u occupies position j ($x_{u,j} = 1$) in a cycle or not ($x_{u,j} = 0$). The 4 different terms of the Hamiltonian describe the following:

1. We want to have cycles which contain each node only once. Fixing a node and running over all the N possible positions in the cycle, the term's contribution is minimized when only one $x_{u,j} = 1$.

2. We do not want multiple nodes to occupy the same position in a given cycle. Following a similar reasoning to the first term, this one's contribution is also minimized when only one $x_{u,j} = 1$.

3. Our graph $G(V, E)$, where V denotes the nodes and E the edges bridging them, has a certain connectivity. Since our path has to use the given connectivity of the graph, this term adds a penalty whenever two consecutive points on a cycle are not connected by an edge.

4. Once we have found a Hamiltonian cycle using the first three terms, we want to minimize the distance travelled. This last term takes into account the weights of the edges $W_{u,v}$ connecting two nodes and promotes constructing a cycle with the minimum distance.

If we are able to find the ground state of the system then, we will have a solution to the problem that satisfies the three different constraints given by the first three terms of H and a minimum travelled distance, given by the forth term.

Since we want to encode this Hamiltonian in a physical quantum system, we will use N^2 qubits, that we will label as $|k\rangle_{iN+j}$, $k = 0, 1$ such that the state of each one will tell us whether node i occupies ($k = 1$) or does not occupy ($k = 0$) position j in the cycle. The $X_{i,j}$ operator that appears in our Hamiltonian, using this convention, has to act on these states as $X_{i,j}\,|k\rangle_{iN+j} = k\,|k\rangle_{iN+j}$, and so we will express it in terms of Pauli operators acting on the qubit $iN + j$ as $X_{i,j} = \frac{1}{2}(1 - Z(iN + j))$.

3 Adiabatic Quantum Optimization (AQO)

The first way in which we will try to find the ground state of the Hamiltonian showed before is by starting off with the following simple Hamiltonian:

$$H_{ini} = \sum_{i=1}^{N^2} -X(i) \tag{2}$$

$X(i)$ now being a Pauli X operator, not out $X_{i,j}$ operator. This Hamiltonian has an easy to compute ground state composed of an equal superposition of all the possible states of our N^2 qubit system.
If we now construct:

$$H(s) = sH_{ini} + (T - s)H_{tsp} \tag{3}$$

And we slowly enough evolve s from 0 to T, starting off in the ground state of H_{ini}, by means of the adiabatic theorem [1] the system will remain in the ground state throughout its evolution, effectively being in the ground state of H_{tsp} which encodes the solution at the end.

4 Quantum Alternating Operator Ansatz (QAOA)

In contrast to AQO, which is an analog approach, as it is not gate-based, the QAOA algorithm is digital, and it is based on alternating so called phase and mixer operators, which depend on a set of parameters to be optimized. The objective of this sequence of parametrized gates/operators is to approximate the adiabatic evolution path that is followed in AQO in a discrete or trotterized way. The two operators that are used take the form:

$$U_P(\gamma) = e^{-i\gamma H_{tsp}} \quad ; \quad U_M(\beta) = e^{-i\beta H_M} \tag{4}$$

Where H_{tsp} is the Hamiltonian that describes the travelling salesman problem that has been already showcased in the first section, and H_M is a mixing Hamiltonian that allows transitions between pairs of states inside the feasable solution's subspace. The mixing Hamiltonian's selection is not unique, and it can be tailored to each problem so that it respects the desired constraints.

By starting with an equal superposition of all states and applying layers of U_P and U_M operators, one can construct a state which depends on the parameters γ and β, which can be optimized classically to get a circuit that outputs the solution to the TSP.

5 Three-node graph

Once we have established how are we going to describe the system and which methods we will use to find solutions, we can move on to to trying to solve the problem for a few simple example cases. The first one will be a three-node fully connected graph with different weights for each edge, shown in figure 1. In this

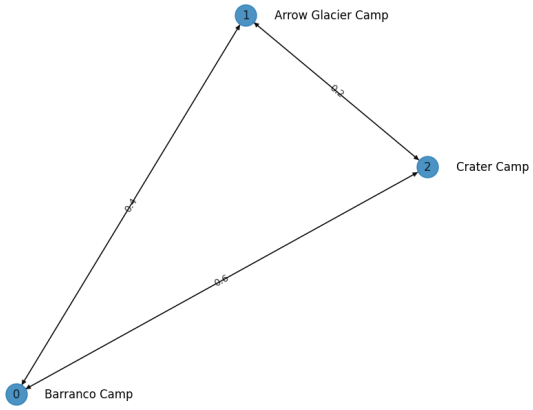

Figure 1: Three node graph, representing a network of base camps located at Kibo Peak, Tanzania.

case we will have $N = 3$ and therefore we will use 9 qubits to represent the 9 degrees of freedom. Since this is a very simple graph, we can already see by just looking at the graph that the optimal path will be to go from node 0 to node 1 and from node 1 to node 2 (or the inverse).

5.1 Three-node graph results using AQO

By applying the procedure showed in section 3 to this graph we get the solution displayed in figure 2, which coincides with the solution one can determine by just looking at the graph, therefore confirming that H_{ini} and H_{tsp} have been correctly constructed.

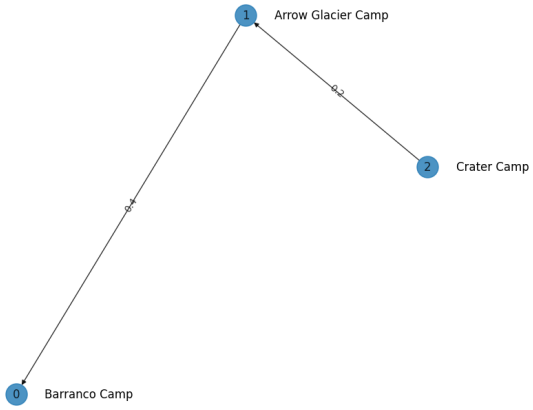

Figure 2: TSP solution to the three-node graph first displayed in figure 1

5.2 Three-node graph results using QAOA

In contrast to the AQO process, for the QAOA, before running any algorithm, we have to choose the mixer Hamiltonian. Following [2] we will use the following Hamiltonian:

$$H_M = \sum_{u<v} \prod_{j=1}^{N} (X_{u,j} X_{v,j} + Y_{u,j} Y_{v,j}) \tag{5}$$

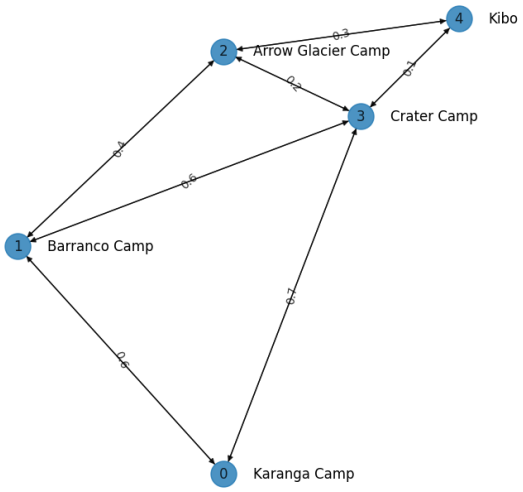

Figure 3: Five-node graph that extends the 3-node graph shown in figure1 with 2 more nodes, representing the starting and initial points. This graph is a representation of the 4 base camps that one must go trough to ascend to mount Kibo.

Apart from this mixing Hamiltonian, another one based on ladder operators, showcased also in [2] was tested, but ended getting solutions which didn't even respect the Hamiltonian cycle constraints. Using this approach, unlike with AQO, the results are not quite so good getting only the correct path only 50% of the time, approximately, which could be a hint that our mixing Hamiltonian is not suitable for our case, since we already know that H_{tsp} does encode correctly the system's information from section5.1.

6 Five-node real life challenge

In this second problem the situation is quite a bit more complex, since we will be working on a 5 node non-fully connected graph, shown in figure 3, with fixed starting and ending points (therefore not looking for a closed loop) and with only 6 qubits available to encode degrees of freedom.
With these restrictions then, the first step is to find the real degrees of freedom of the problem to go from 25 to 6. By analyzing the graph we can see that:

1. The first and last nodes are fixed. Therefore we only have to work with 3 camps, reducing the number of degrees of freedom from 25 to 9.

2. In the 5-position path, node 1 can't occupy position 4 and node 2 can't occupy position 2 due to the connectivity of the graph. The degrees of freedom then drop from 9 to 6.

3. While we are not looking at closed loops, we still do not want to step into the same node twice in our path. If we look at the graph, we can see that if node 3 is (is not) in position 2 on the path, node 1 must be (not be) in position 3, since there are no edges connecting it directly to node 4. We have then that $x_{1,2} = x_{3,1}$, reducing the number of real degrees of freedom to 6, which is the amount that we can work with with our 6-qubit limitation.

With this in mind, and since this approach is very tailored to the graph shown in figure 3, we construct the

following hamiltonian by hand:

$$\begin{aligned} H = &(1 - x_{1,2} - x_{1,3})^2 + (1 - x_{2,3} - x_{2,4})^2 + (1 - x_{3,2} - x_{3,3} - x_{3,4})^2 \\ &+(1 - x_{1,2} - x_{3,2})^2 + (1 - x_{1,3} - x_{2,3} - x_{3,3})^2 + (1 - x_{2,4} - x_{3,4})^2 \\ &+W_{0,1}x_{1,2} + W_{1,3}x_{1,2}x_{3,3} + W_{3,2}x_{3,3}x_{2,4} + W_{2,4}x_{2,4} \\ &+W_{0,1}x_{1,2} + W_{1,3}x_{1,2}x_{2,3} + W_{2,3}x_{2,3}x_{3,4} + W_{3,4}x_{3,4} \\ &+W_{0,3}x_{3,2} + W_{3,1}x_{3,2}x_{1,3} + W_{1,2}x_{1,3}x_{2,4} + W_{2,4}x_{2,4} \end{aligned} \tag{6}$$

Where the first row encodes the constraint of each node appearing only once in the cycle, the second row encodes the constraint of each position in the cycle being filled by only one node, and the third to last rows encode the cost function of the only 3 possible cycles that can be formed with the given constraints. Since we have already considered the connectivity of the graph, the third term that appeared in equation 1 is not needed.

Apart from using a tailored Hamiltonian, we will also change how we label the qubits for the sake of simplicity, as now, while we only have 6 qubits, since we are still labelling the nodes from one to three, $iN + j$ would still go up to 16. For this reason, for this part of our work we will work with the following convention:

$x_{i,j}$	$x_{1,2}$	$x_{1,2} = x_{3,2}$	$x_{2,3}$	$x_{2,4}$	$x_{3,3}$	$x_{3,4}$
qubit	0	1	2	3	4	5

Table 1: Assignment of the 6 degrees of freedom for the problem of the shortest path between the nodes 0 and 4 of the graph shown in figure 3.

6.1 Five-node graph results using AQO

Repeating the procedure of 5.1 but using this new Hamiltonian presented with 6 degrees of freedom, we get the path displayed in figure 4, which while not as obvious as in the 3 node graph, it is indeed the shortest path possible between node 0 and node 4, which is a clear sign that our Hamiltonian built upon a few logical reasoning steps does indeed encode all the necessary information to solve our problem while only requiring 6 qubits.

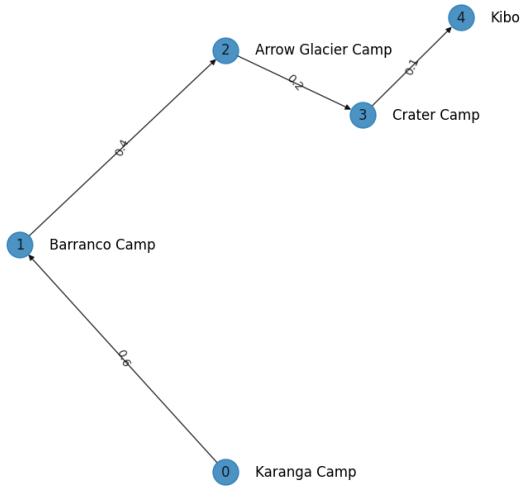

Figure 4: Graph showing the path of shortest distance between node 0 and node 4 for the graph displayed in 3.

6.2 Five-node graph results using QAOA

In contrast to the 3-node case, this time we do not have a general 5-node graph and therefore, the construction of the mixing Hamiltonian given in [2] is not so straight forward as $x_{i,j}$ is not associated anymore with $|k\rangle_{iN+j}$. To build the same XY Hamiltonian we will use the same construction as in equation 5 but imposing that both indexes (u, j) and (v, j) appear in the allowed values showed in table 1, and then mapping those indexes directly into the respective qubits.

With this approach, the QAOA algorithm does return the same right answer we got using the AQO for this restricted case, displayed in figure 4, but now with a much higher probability than in the 3 node general case, probably due to the fact that the feasable solution's subspace is much smaller.

All in all, having both algorithms return the same answer, one that is indeed correct, we can conclude that our Hamiltonian, while not very elegant in construction, is indeed effective at describing the system while using a reduced amount of resources.

References

[1] T. Albash and D. A. Lidar, "Adiabatic quantum computation," *Reviews of Modern Physics*, vol. 90, jan 2018.

[2] S. Hadfield, Z. Wang, E. G. Rieffel, B. O'Gorman, D. Venturelli, and R. Biswas, "Quantum approximate optimization with hard and soft constraints," *Proceedings of the Second International Workshop on Post Moores Era Supercomputing*, 2017.