Olivia Pitcl

4 May 2022

Phys 305, K Johns

Creative Project Summary

For my creative project, I solved the Traveling Salesman Problem (TSP) using Classical and Quantum computing. The Classical component was done using the nearest neighbor algorithm, which is an heuristic and therefore incomplete solution, though it is much faster than the brute force algorithm which is a complete solution. When running the code, if you print out the costs of each path you should expect to see the nearest neighbor method find an equal or more costly path than that of the brute force method. The Quantum component was done using the Qiskit Software Development Kit, specifically Qiskit.Aer, which is a simulator package. This package uses a Classical computer to simulate the effects of a Quantum computer to perform the algorithm. The cost of this program should give the same output as that of the brute force algorithm.

Each algorithm was coded in entirely different manners. The result using the nearest neighbor method is found by looping through all possible costs to travel from a given node to its neighbors, choosing the nearest neighbor, and then doing the same for that node until we reach the end. I did this using two for loops, which means the runtime will depend on $n(n+1)$, or $O(n^2)$. The brute force method calculates every single cost of every single path. It was done by feeding in a matrix of node costs like those in the example data files I submitted and whether they have been visited. You can choose to use those data files for the code or randomized matrices. Using these data structures, the code uses recursive backtracking to update the minimum value as it calculates each cost.

The Quantum method employs a method very similar to that of the brute force method. It calculates every path and their cost. Instead of a matrix of costs as its input, however, the circuit builds each path based on every Hamiltonian cycle (which we must find/hard code as a series of one's and zero's) and calculates the cost of the path by building a wavefunction with the node-to-node cost encoded in the net phase of the wavefunction, normalized between zero and two pi. The total cost is arranged by measuring/collapsing the resulting wave functions and turning each measurement into a classical bit (a one or a zero). Having each cost, we search through and find the minimum using Grover's algorithm. I wanted to compare the costs to one another, but when the phase of the costs became too abnormal (i.e. factors of pi we do not typically use in math classes), the cost measurements began to vary. To find the correct cost, I had to run the code multiple times and use the majority as the cost for a given cycle of nodes. Perhaps for 4-node problems I can map the cost of each phase to some fraction of pi we can easily calculate. This becomes tricky, however, for problems with more cities.

In this problem, I took the powers of a future technology and forced it into the realm of the classical in an attempt to see how they would measure up against one another. Overall, this was quite an exciting project for me. I have always wanted to try my hand at the Qiskit SDK and this was the perfect opportunity. I gained an entry-level understanding of the way a Quantum Circuit works and learned two algorithms with them. One of the things I love about Quantum Computing is that it provides all these new computational abilities, which we saw in this problem through our capacity to use Grover's algorithm, which I have never been exposed to before. Thank you for the opportunity to do this project, and for a good semester overall.