The Travelling Salesman Problem with Qiskit Runtime API

Jaimil Dalwadi, Freya Shah, Yoyo Liu, Abiskar Thapa Kshetri

IEEE-Services Quantum Hackathon Submission

August 30, 2021

Introduction:

Quantum computing with quantum algorithms have been used to optimize classical algorithms. By replacing classical bits with quantum bits, it is possible to speed up the calculation process and create better output results. In this program we created, we have used Qiskit Runtime's built-in library to speed up and optimize the complex travelling salesman problem with 4 nodes. Our approach is to implement the Variational Quantum Eigensolvers (VQE) Program from Qiskit Runtime to run the solution and also test the Direct API Calls. We hope by doing this, the shortest path problem can be enhanced and be prepared for future application.

1. Motivation:

The motivation of this project comes from the merging and development of artificial intelligence and quantum computing. The creation of the Dijsktra's algorithm for the shortest path problem has been a huge help in many fields, including in the development of robotics. With the help of quantum computing, we aim to optimize the solution while providing a faster method. We hope to facilitate the use of quantum computing and introduce the shortest path solver in Qiskit Runtime. Our passion for quantum computing is our motivation, and our collective desire to improve and share our project with the community is the reason we think Qiskit Runtime library can be a great fit for our project. We hope that in the future, our project could be uploaded to the Qiskit Runtime, so it is open source and accessible to everyone.

2. Innovation:

This project includes a Qiskit Runtime routine to solve the travelling salesman problem with 4 nodes. As of now, there are tutorials and example solutions to solve this problem using the local VQE object from qiskit_optimization, however, there isn't a solution that solves this problem with 4 nodes specifically through an API call. We wanted to test this out as it hasn't been done before and see what we could build. By doing so, we would be one of the first to develop an applicable solution using the Runtime API.

3. Applicability:

As of right now, Qiskit Runtime API is in the beta-testing phase. However, when it is open to the public for uploading and testing programs as well as making direct API calls, this solution can be implemented for practical use. Completing and publishing this solution could take anywhere from a few weeks to a few months but is very feasible for the near-term advancements in quantum computing. To apply this sort of algorithm to real world situations such as Amazon delivery services, drones, and other travel systems, many different requirements would need to be met. The Runtime API must be versatile enough and powerful enough to process several quantum jobs in real-time to avoid crashing and to provide time efficiency. We expect this sort of real-world application to be used within the next 3 years as there has been an exponential speedup in quantum computing applicability during this decade.

4. Role of Qiskit Runtime:

Qiskit Runtime helps streamline computations requiring many inputs. It allows users to upload their Qiskit quantum programs to the cloud infrastructure. This program also known as runtime program can be invoked by same or other authorized users by simply passing in the parameters. For the travelling Salesman Problem, we have used Qiskit Runtime VQE built-in program. This increases the speed of the execution and the cloud service is the first step to bringing quantum computing closer to the practical application stage.

5. Future application:

This program, along with many other shortest path solvers, can be used in many applications. One of the many applications is in AI and robotics. Engineers and AI experts have been using the shortest path algorithms to plan and direct the paths of robots. These classical algorithms enable the development of robotics, and with our project, we hope to optimize the speed, decrease run time, increase accuracy, and add more constraints. Many other future applications can be discovered once our project is fully transferred into Qiskit Runtime. We aim to benefit future researchers and create more applications in the future.

Going forward, we would like to build a program which we can upload to the API for others to use as a TSP solution. That way, users only need to input cities/locations required as well as other constraints without having to worry about building hamiltonians or algorithms. Once public access has been released for uploading custom programs, we plan to work on this program, bringing quantum computing one step closer to the public, accessible, and applicable stage.

6. Technology stack, design decisions and architecture:

We have used Jupyter Notebooks for implementing our code as it displays the output with utmost clarity and is widely used for quantum programs. It allows us to create and share documents with live code, equations, computational output, visualizations, and other resources along with explanatory text in a single document. Jupyter notebook is also compatible with GitHub where all our code is uploaded for users to view.