CSC 591 Project B4 report2
Quantum Enhanced Simulation Optimization

Yunsoo Ha

10/22 2020

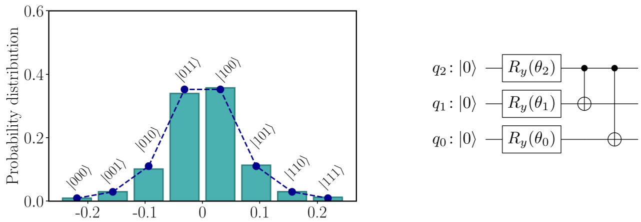
1 Introduction

As we mentioned before, we need to focus three things: Quantum Amplitude Estimation (QAE), Operator $\mathcal{P}_X$, and F. At this report, we will focus on Operator $\mathcal{P}_X$, and F and show simple example.

2 Operators

2.1 Operator $\mathcal{P}_X$

Firstly, we need to load the probability density function by using n-qubits. As we have learned from the lecture, we can load the bell-shaped probability distribution like normal distribution easily by using $CNOT$ and R_y (Figure 1).



(a) (b)

Figure 1: 3-qubit bell-shaped probability distribution

However, we cannot always easily load an exact representation of a generic data structure into an n-qubit state. Hence, the paper suggested qGAN which can learn a representation of the probability distribution underlying the data samples and load it into a quantum state. By using this method, even if it does not have closed-form probability density function, we can load any probability distribution. Qiskit already have libraries for qGAN and loading general probaility distribution such as normal, log-normal distribution. Since we will focus on QAE and simulation optimization, we don't deal with this method in detail.

2.2 Operator F

We can make the circuit for the following operator F:

$$F|x\rangle|0\rangle = \sqrt{1-f(x)}\,|x\rangle|0\rangle + \sqrt{f(x)}\,|x\rangle|1\rangle$$

For F, we can use Multi-controlled $R_y(2\theta)$ (figure 2).

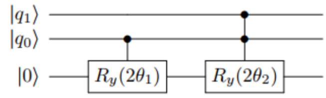


Figure 2: 3-qubit circuit for the polynomial function

Then, we can express simple polynomial function as follows (figure 2).

$$\begin{aligned} |q_1\rangle|q_0\rangle|0\rangle &\to |q_1\rangle|q_0\rangle e^{-i\theta_2 q_0 q_1 Y} e^{-i\theta_1 q_o Y}|0\rangle & (1)\\ &= |q_1\rangle|q_0\rangle(\cos(\cdots)|0\rangle + \sin(\theta_2 q_0 q_1 + \theta_1 q_0)|1\rangle) & (2)\\ &= |q_1\rangle|q_0\rangle(\cos(p(q))|0\rangle + \sin(p(q))|1\rangle) & (3) \end{aligned}$$

,where $p(q) = \theta_2 q_0 q_1 + \theta_1 q_0$ is a polynomial.

Here, we need one more technique. Exploiting the near linearity of the sine function when the argument is close to 0, we prepare instead $cp(q)$. Then for $c \in$ R small enough we have that

$$|q_1\rangle|q_0\rangle(\cos(\cdots)|0\rangle + \sin(cp(q))|1\rangle) \approx |q_1\rangle|q_0\rangle(\cos(\cdots)|0\rangle + cp(q)|1\rangle)$$

3 Example

In this section, we introduce simple example with 2-qubits and $f(x) = x^2$, where $x \in \{0, 1, 2, 3\}$. First, we prepare following bell-shaped probability distribution.

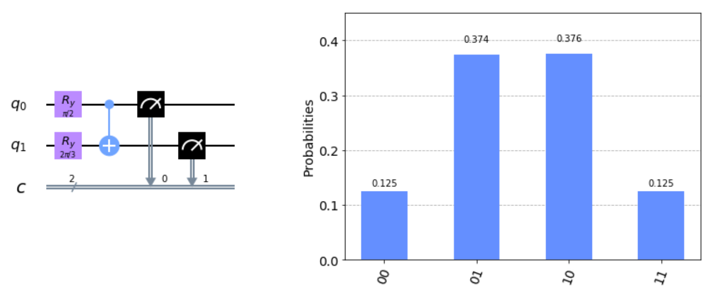


Figure 3: 2-qubit bell-shaped probability distribution

Now, we want to apply function $f(x) = x^2$. Thus, we look for the circuit for $f(x) = x^2$, where $x \in \{0, 1, 2, 3\}$ is represented by two qubits. In this case, we have for $q_i \in \{0, 1\}$, $x = 2q_1 + q_0$. Hence,

$$f(x) = f(2q_1 + q_0) = (2q_1 + q_0)^2 = 4q_1 + q_0 + 4q_1q_0$$

Hence, we can make a circuit for above equation by using Multi-controlled $R_y(2\theta)$ (figure 4).

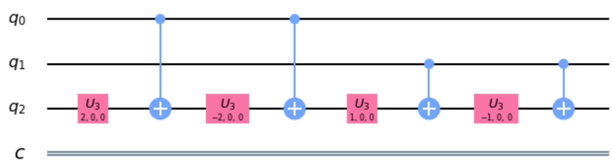


Figure 4: The circuit for $f(x) = x^2$

As a result, we can express the circuit (figure 5) for

$$\sum_{i=0}^{3} \sqrt{1 - cf(i)}\sqrt{p_i} \left|i\right\rangle_2 \left|0\right\rangle + \sum_{i=0}^{3} \sqrt{cf(i)}\sqrt{p_i} \left|i\right\rangle_2 \left|1\right\rangle$$

,where c = 0.001.

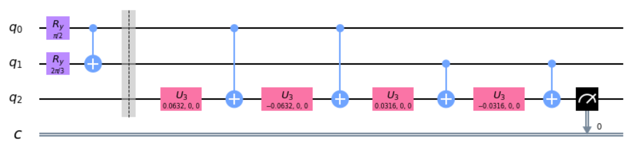


Figure 5: The entire circuit

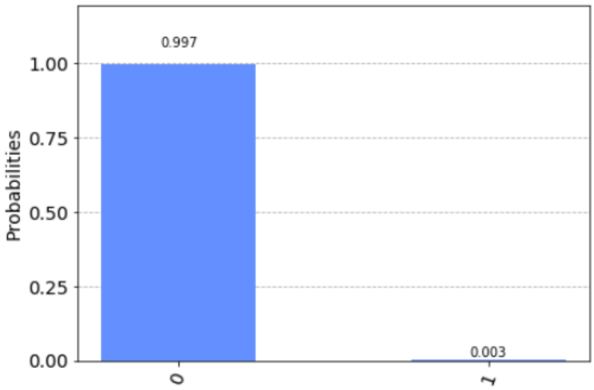
It means when we measure the last qubit, we can get the probability

$$P(1) = \sum_{i=0}^{3} cf(i)p_i = E(cf(x)) = cE(f(x))$$

,where x follows probability distribution p_i (figure 3) and $x \in \{0, 1, 2, 3\}$. Thus, we can get the $E(f(x))$. If we calculate the exact value for $E(x)$, $E(x) = 0 \times 0.125 + 1 \times 0.375 + 4 \times 0.375 + 9 \times 0.125 = 3$. If we use our circuit, then we can get 2.97 (figure 7).

Now, we can see our $P(1)$ is too small because of $c = 0.001$. Hence, we need QAE based on grover algorithm to get the quantum advantage. Entire code is uploaded at Github (https://github.ncsu.edu/yha3/CSC591).

```
backend = BasicAer.get_backend('qasm_simulator')
shots = 100000
results = execute(qo, backend=backend, shots=shots).result()
answer = results.get_counts()
plot_histogram(answer)
```




```
probability = answer['1']/shots
Approximate_answer = probability/cost
```

```
print('Exact value:    \t%.4f' % exact_answer)
print('Estimated value:\t%.4f' % Approximate_answer)
```

```
Exact value:        3.0024
Estimated value:    2.9700
```

Figure 6: The result

4 Time line

Date	Description
2020/10/15	Implement the operator $\mathcal{P}_X$ and do the research related to it
2020/10/22	Make the circuit for the operator F (for any function $f(x)$)
2020/10/29	Implement various QAE without QPE and try to improve it
2020/11/05	Implement the Figure 1(b) circuit for entire quantum SO
2020/11/12	Use entire algorithm for an application problem
2020/11/19	Prepare the presentation

Reference

Woerner, Stefan, and Daniel J. Egger. "Quantum risk analysis." npj Quantum Information 5.1 (2019): 1-8.

Grinko, Dmitry, et al. "Iterative quantum amplitude estimation." arXiv preprint arXiv:1912.05559 (2019).

Brown, Eric G., Oktay Goktas, and W. K. Tham. ”Quantum Amplitude Estimation in the Presence of Noise.” arXiv preprint arXiv:2006.14145 (2020).

Suzuki, Yohichi, et al. ”Amplitude estimation without phase estimation.” Quantum Information Processing 19.2 (2020): 75.

Zoufal, Christa, Aurélien Lucchi, and Stefan Woerner. ”Quantum generative adversarial networks for learning and loading random distributions.” npj Quantum Information 5.1 (2019): 1-9.

Gacon, Julien, Christa Zoufal, and Stefan Woerner. ”Quantum-Enhanced Simulation-Based Optimization.” arXiv preprint arXiv:2005.10780 (2020).

Vázquez, Almudena Carrera. Quantum Algorithm for Solving Tri-Diagonal Linear Systems of Equations. Diss. ETH Zürich, 2018.

URL: https://github.ncsu.edu/yha3/CSC591