Quantum Walk

B2 AQUA
cocori
Faculty:Rodney D. Van Meter, Takahiko Satoh

Background

- We(parton, rum, me) are adopted for the mitou target project of gate quantum computer.
- We think quantum walk can be a solution of our suggestion.
- I want to expand it to use for larger problem.

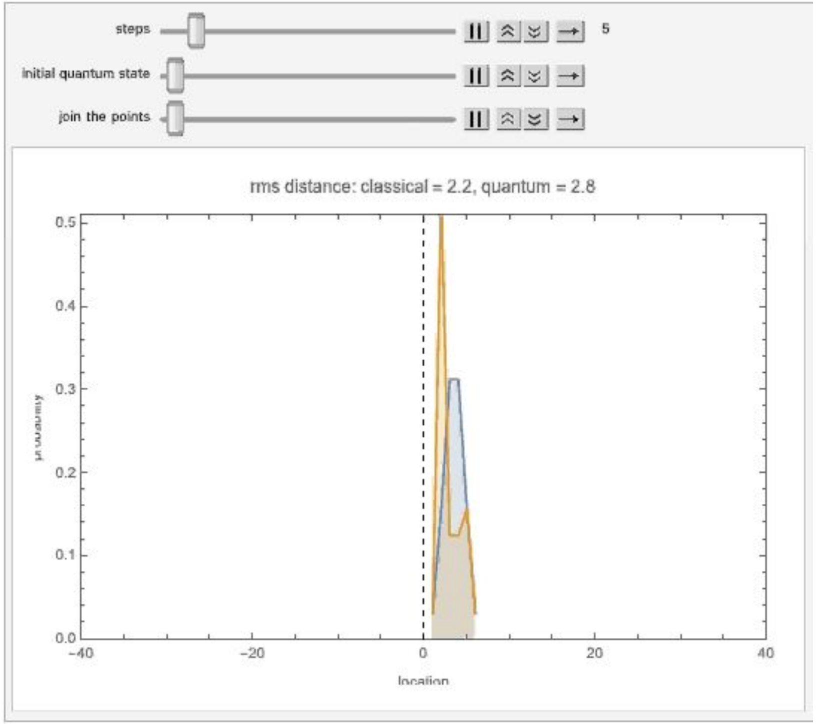
MITOU 未踏ＩＴ人材発掘・育成

MITOU ADVANCED 未踏的ＩＴ起業家・事業家育成

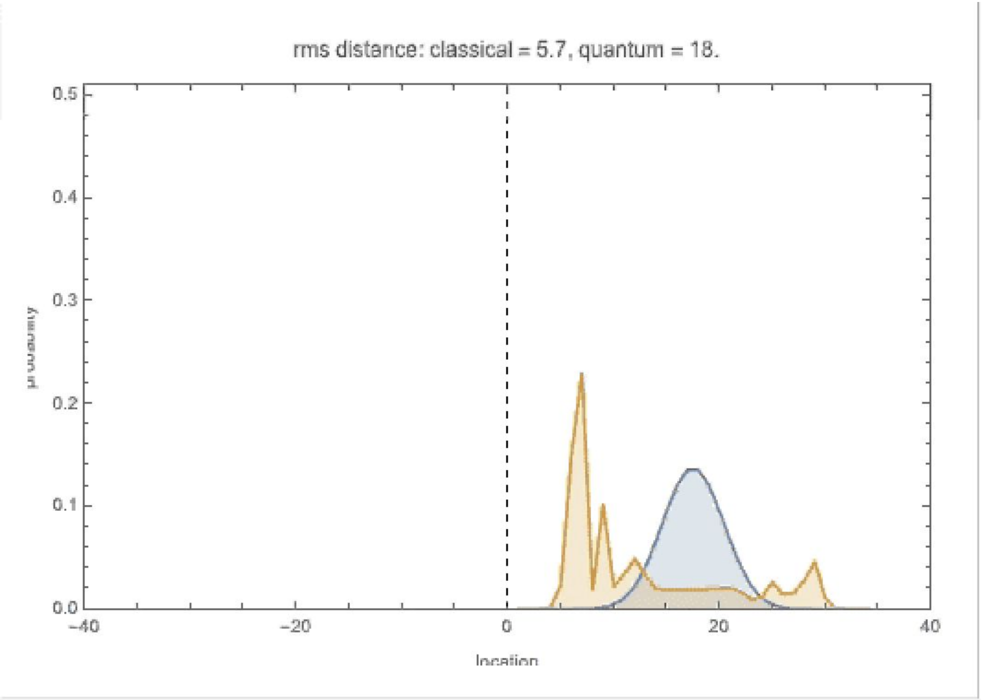
MITOU TARGET 次世代先端ＩＴ技術者育成

https://www.ipa.go.jp/jinzai/target/2018/koubokekka2_index.html

Overview (What is quantum walk?)



Mathematica Simulator

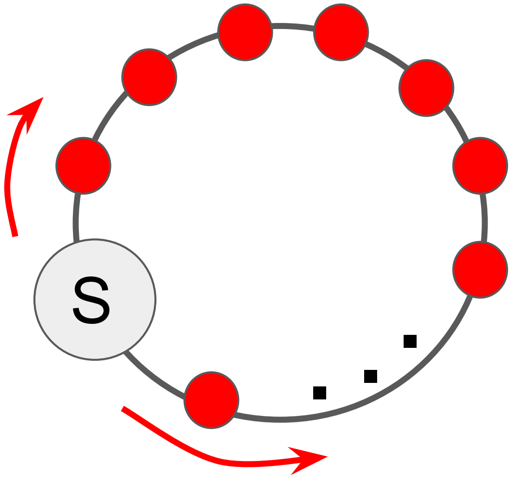


ref:http://demonstrations.wolfram.com/QuantumRandomWalk/

Implementation

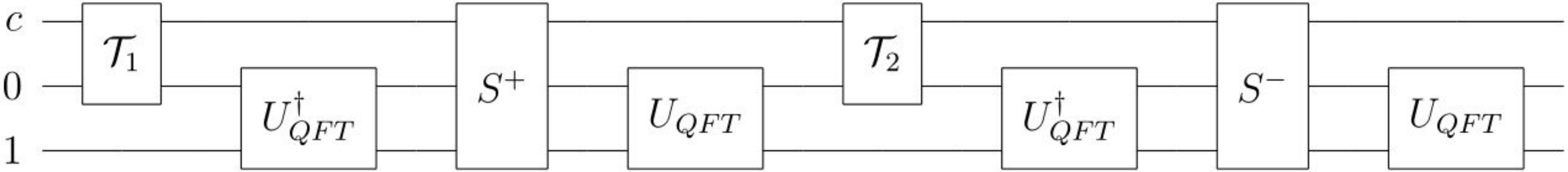
Language	Python(3.6.5)
Module	Qiskit(0.6.1), matplotlib...
Device	ibmq_20_tokyo ibmqx_hpc_qasm_simulator

Implementation



I referred to [1] and implemented the quantum walk on the lattice of circle.

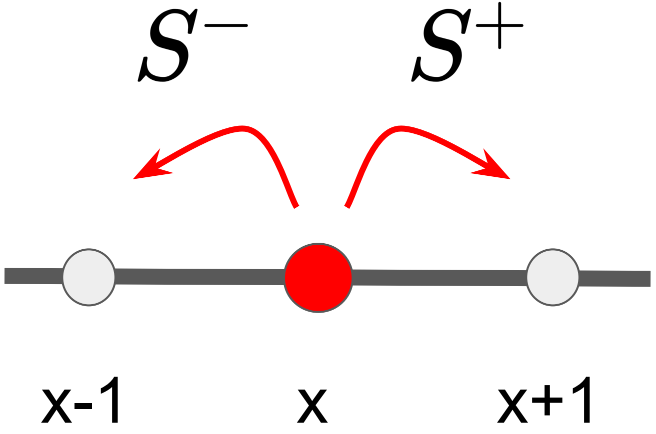
[Radhakrishnan Balu, Daniel Castillo, George Siopsis(2017) arXiv:1710.03615 [quant-ph]]



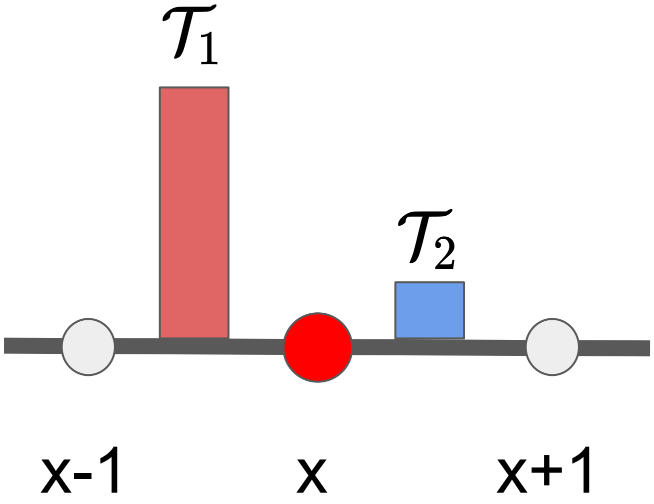
Expand from 1 step to any steps.

Implementation

Step operators $S^{\pm}$



Coin operators $(\mathcal{T}_1, \mathcal{T}_2)$



Implementation

Quantum Fourier Transformation

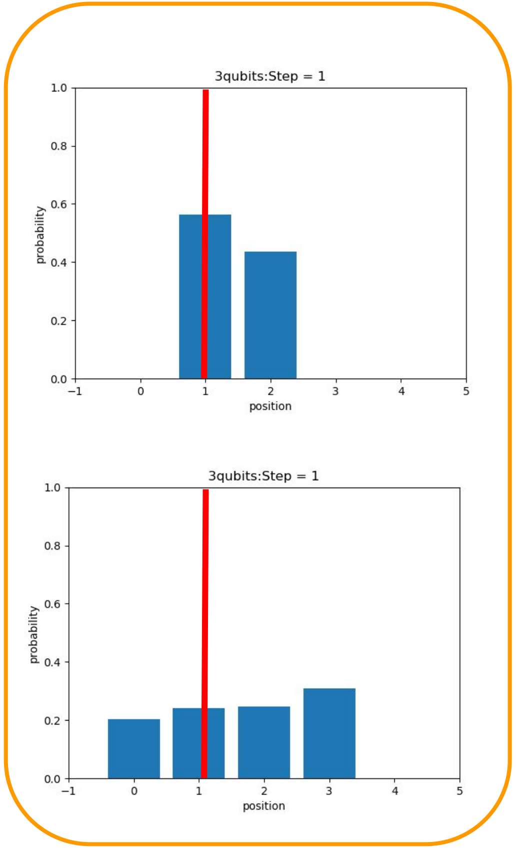
```python
def _QFT_dg(self, circuit):
    q = self.q
    qc = circuit
    for n in range(1, self.qubits):
        # print(n)
        qc.h(q[n])
        for t in range(1, self.qubits-n):
            try:
                qc.cu1(pi/(2**(t)), q[n], q[n+t])
            except:
                continue

def _QFT(self, circuit):
    qc: QuantumCircuit
    qc = circuit
    for m in range(self.qubits-1, 0, -1):
        for u in range(self.qubits, 0, -1):
            try:
                qc.cu1(-pi/(2**(u)), q[m+u], q[m])
            except:
                continue
        qc.h(q[m])
```

Total walk

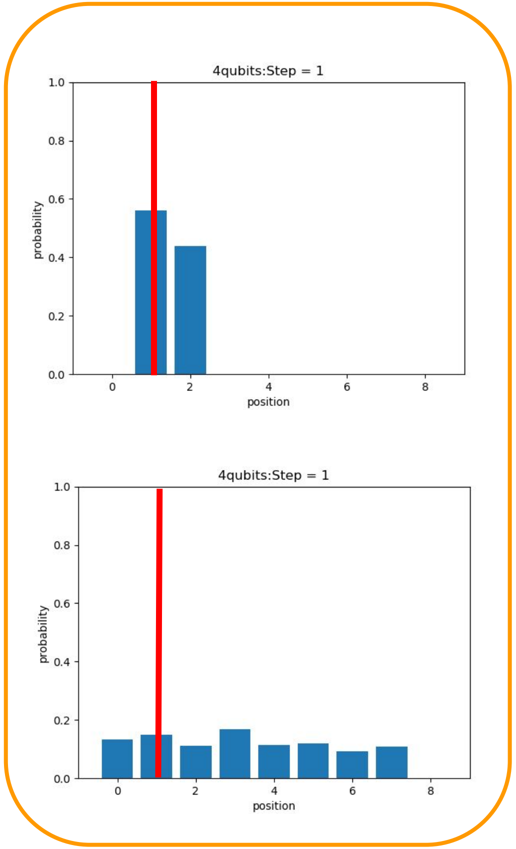
```python
def walk(self, step):
    c = self.c
    q = self.q
    qc = self.qc

    # qc.x(q[1])
    qc.x(q[self.qubits-1])
    # initial coin perator
    for i in range(step):
        self._coin_1(-1/8, -1/8, qc)
        # self.check_qft_dg()
        self._QFT_dg(qc)
        self._S_plus(qc)
        self._QFT(qc)
    for j in range(step):
        # initial coin operator2
        self._coin_2(-1/8-pi/2, 1/8+pi/2, qc)
        self._QFT_dg(qc)
        self._S_minus(qc)
        self._QFT(qc)

    for i in range(self.qubits):
        qc.barrier(q[i])

    mq = [j for j in range(self.qubits-1, 0, -1)]
    mc = [k for k in range(self.qubits-1)]
    for j, k in zip(mq, mc):
        qc.measure(q[j], c[k])
```

Evaluation

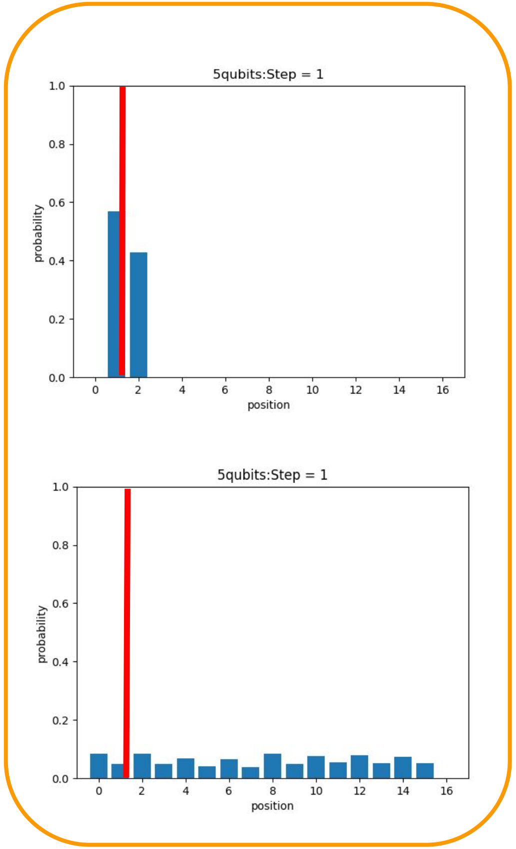
3qubit



4qubit

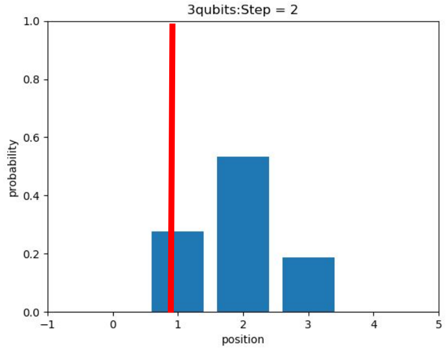


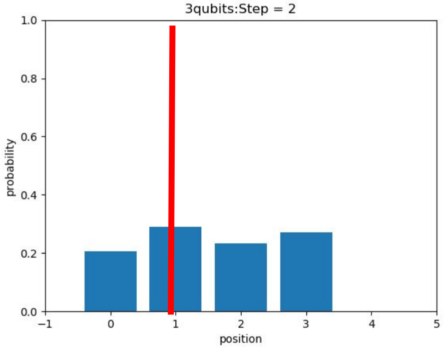
5qubit



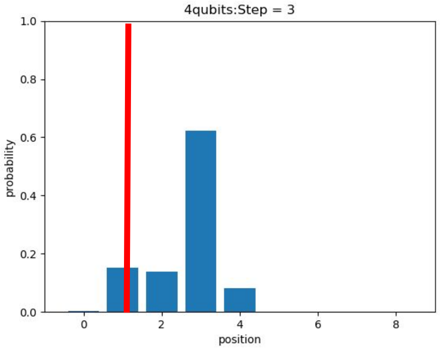
Evaluation

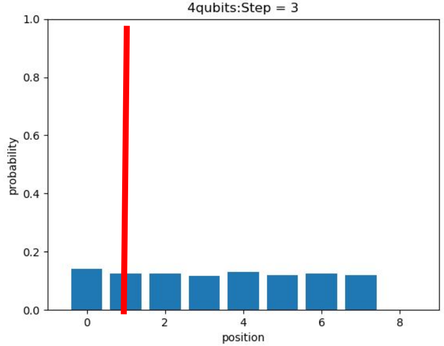
3qubit



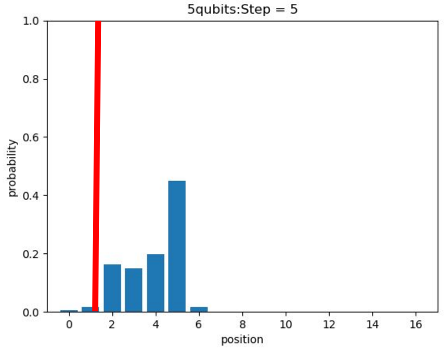


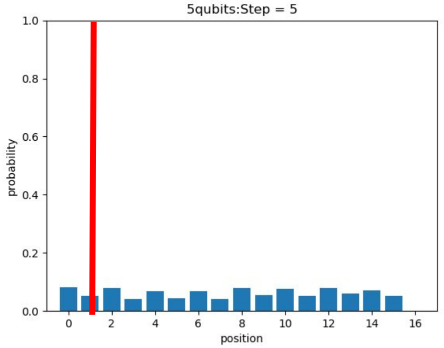
4qubit





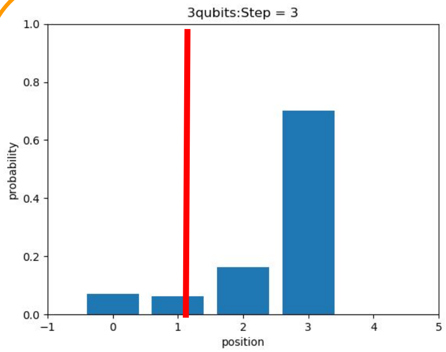
5qubit

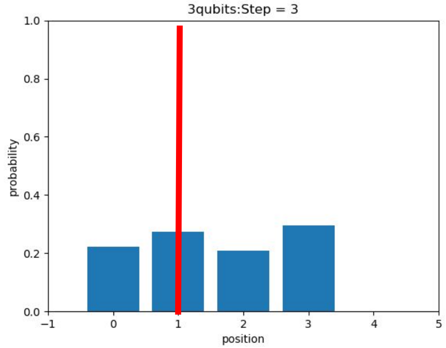




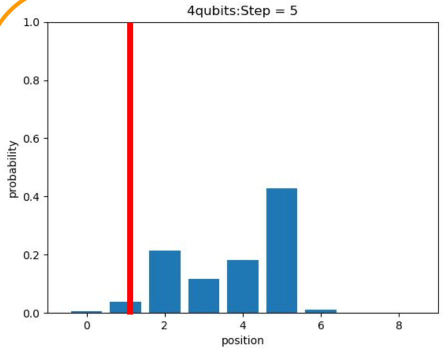
Evaluation

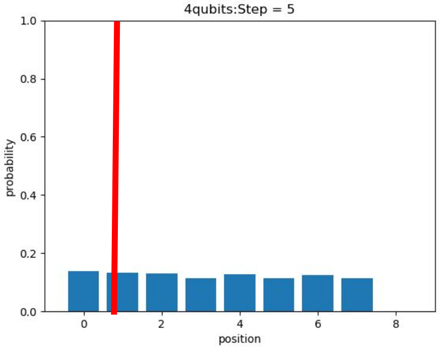
3qubit



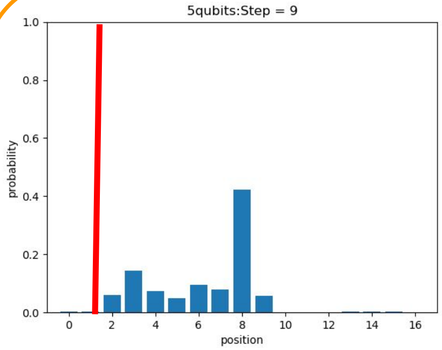


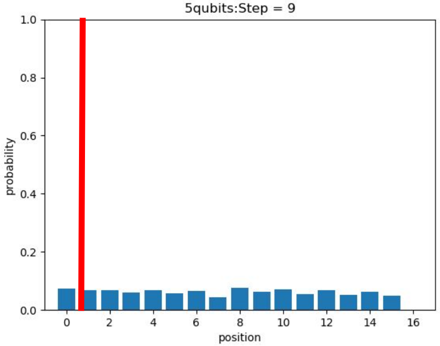
4qubit





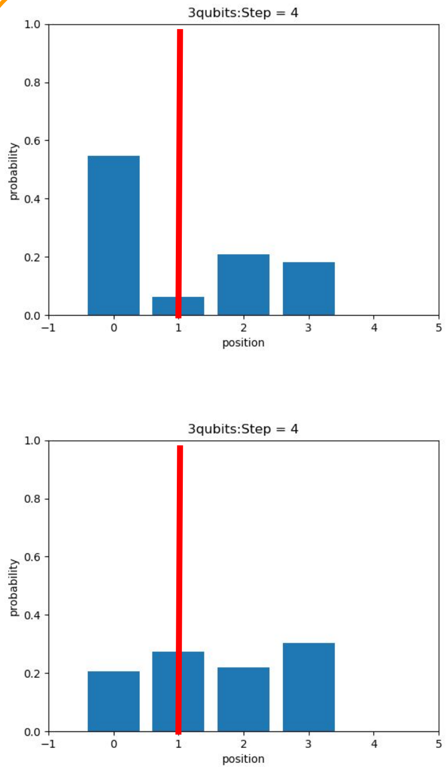
5qubit



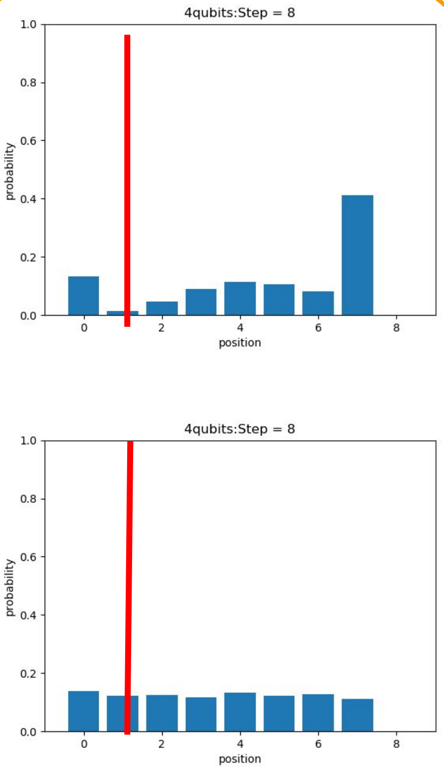


Evaluation

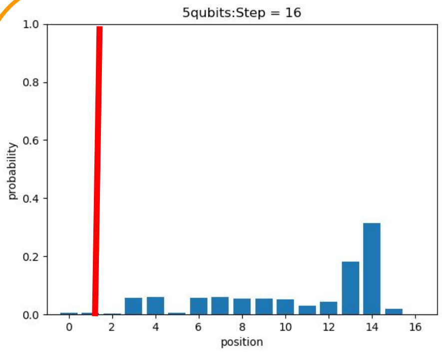
3qubit



4qubit



5qubit



No data

Error consideration

$$Success\,rate \approx 0.95^n$$

n...the number of two qubit gate.

3qubits 1step	n = 20	$0.95^{20} = 0.35$
4qubits 1step	n = 32	$0.95^{32} = 0.19$
5qubits 1step	n = 52	$0.95^{52} = 0.069$

Conclusion

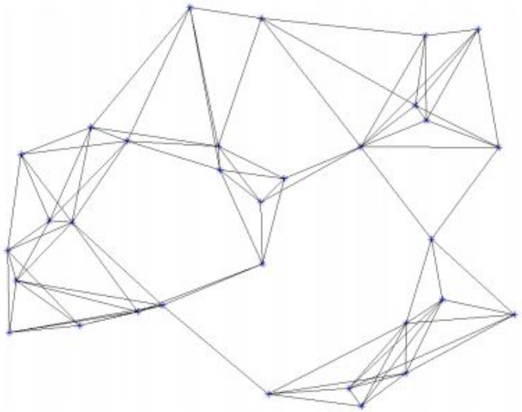
- I could check the **expansion of probability distribution** in hpc simulator.
- In the **NISQ** device, it does not work well.
- If we want to apply some real problems, we have to use **error correction** or **divide the problems** that we can simulate.

Future Work

- Apply the quantum walk to graph kernels.

There is the way of using classical random walk.(Smola A.J., Kondor R. (2003))

→Can I apply the quantum walk to graph kernel?



A nearest neighbor graph.

Reference

[1]Radhakrishnan Balu, Daniel Castillo, George Siopsis(2017) Physical realization of topological quantum walks on IBM-Q and beyond arXiv:1710.03615 [quant-ph]

[2]Matthew Falk(2013). Quantum Search on the Spatial Grid arXiv:1303.4127 [quant-ph]

[3]今野　紀雄(2014). 量子ウォーク　森北出版

[4]Yaakov S. Weinstein(2014). Quantum error correction during 50 gates Phys. Rev. A 89, 020301(R)

[5]Takuya Kitagawa, Mark S. Rudner, Erez Berg, Eugene Demler(2010) Exploring Topological Phases With Quantum Walks Phys. Rev. A 82, 033429