Quantum Research Project

Pierriccardo Olivieri

Politecnico di Milano

August 3, 2020

outline

- Random/Quantum walk intro
- Coined DTQW in cycle graph
- Results
- Generalization
- Performance
- Szegedy Quantum Walks
- Applications
- Conclusions

What is a Random walk?

Definition (Random walk)

Is a mathematical object that describe a random path over a mathematical space

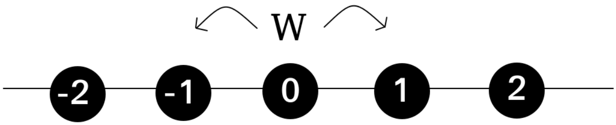
We can identify two type of Random Walk

- discrete time
- continuous time

Random Walk on a line

Example (Random walk on a line)

- line of integer numbers $\mathbb{Z}$
- walker position
- random experiment
- move the walker



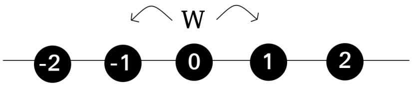
Quantum Walks

Analogous of random walks, again the focus is on Discrete time quantum walks

Example (quantum walk on a line)

To construct a Quantum walk on a line we need

- mathematical space
- position
- coin operator
- shift operator
- state of the system



Mathematical space

The space in which the Coined QW is defined is defined by the Hilbert space

$$\mathcal{H} = \mathcal{H}_p \otimes \mathcal{H}_c \tag{1}$$

where $\mathcal{H}_p$ and $\mathcal{H}_p$ are the Hilbert spaces of the coin and position

Position

Position identified by vector:

$$|position\rangle \in \mathcal{H}_p \tag{2}$$

encoding in binary labels we need Log(N) qubits to represent the number.

Example ($N < 8 => 3qubits$)

- $0- > 000- > |000\rangle$
- $1- > 100- > |100\rangle$
- $7- > 111- > |111\rangle$

Coin operator & state of the system

The **coin operator** is a vector in a 2-dimensional Hilbert space

$$|coin\rangle \in \mathcal{H}_c \quad where \mathcal{H}_c = |0\rangle, |1\rangle \tag{3}$$

Combining element defined before a **state** of the system is:

$$|\phi_{initial}\rangle = |position\rangle_{initial} \otimes |coin\rangle_{initial} \tag{4}$$

Shift operator

The operator that actually perform the shift of the walker depending on the outcome of the coin

$$S = |0\rangle_c \langle 0| \otimes \sum i |i+1\rangle_p \langle i| + |1\rangle_c \langle 1| \otimes \sum i |i-1\rangle_p \langle i| \tag{5}$$

Coined quantum walk operator

Combining the element defined before we obtain the operator that perform one step of the quantum walk:

$$U = Sx(C \otimes I_p) = SC \tag{6}$$

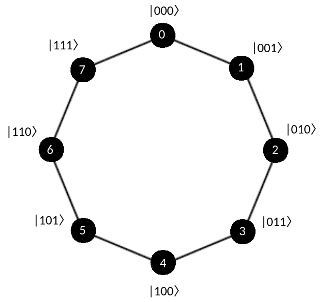
Coined DTQW on a cycle graph

Now we want to implement the Coined DTQW on a cycle graph with N = 8

Circuit:

- position encoded in 3 qubits e.g. $7->|111\rangle$
- coin operator: Hadamard coin (1 qubit)
- shift operator $|i+1\rangle$ or $|i-1\rangle$

Cycle Graph C_8



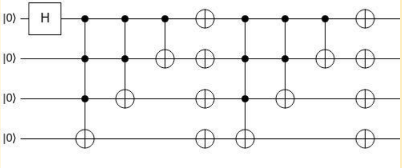
Circuit implementation

We need 3 element to construct the circuit:

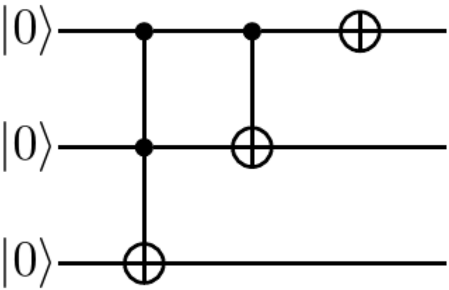
Circuit Components

- increment circuit
- decrement circuit
- Hadamard gate

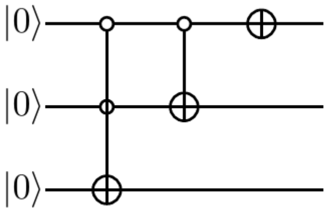
Final Circuit

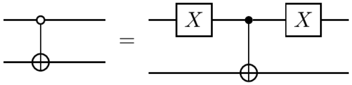


Increment circuit

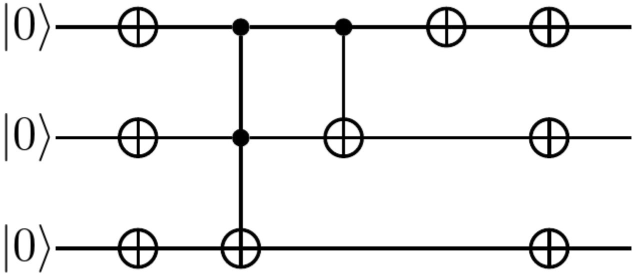


Decrement circuit

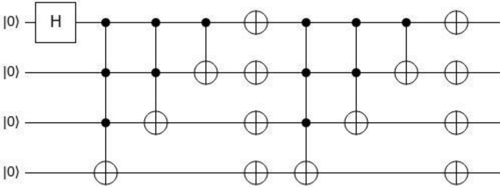


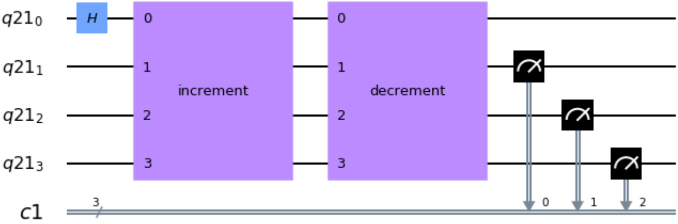


Decrement circuit



Complete circuit





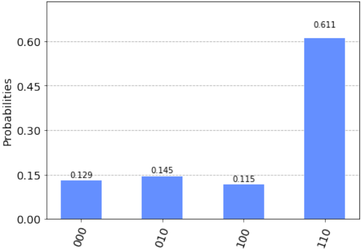
Click here for Quirk Simulation

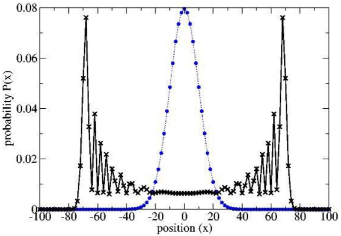
Results

To perform a Quantum walk we apply the circuit several times starting from a certain position

Results

- true randomness behavior (vs classic gaussian)
- probability of measure odd number starting from an even is close to zero
- Hadamard coin treats two direction in different way





Generalization

Two Generalization analysis

w.r.t. model

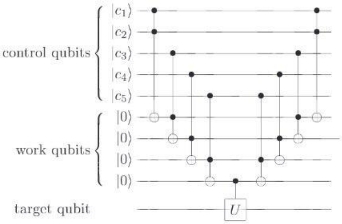
- different type of coin (or model without coin)
- different type of shift operator
- Generalization for dimensions ¿ 1

w.r.t. to class graph

- increasing nodes number we need ancillary qubits
- each graph/graph class needs a specific circuit

Multi controlled gates

- are difficult to implement
- implementation may require ancillary qubits
- various type (ancillary, rotations)
- can lead to inefficiency of the circuit




```
def mcx(self, control_qubits, target_qubit, ancilla_qubits=None, mode='noancilla'):
    """Apply :class:`~qiskit.circuit.library.standard_gates.MCXGate`.

    The multi-cX gate can be implemented using different techniques, which use different numbers
    of ancilla qubits and have varying circuit depth. These modes are:
    - 'no-ancilla': Requires 0 ancilla qubits.
    - 'recursion': Requires 1 ancilla qubit if more than 4 controls are used, otherwise 0.
    - 'v-chain': Requires 2 less ancillas than the number of control qubits.
    - 'v-chain-dirty': Same as for the clean ancillas (but the circuit will be longer).
    """
```

Performance

Definition (Search problem)

find a marked vertex in a graph using Quantum Walks and measure where the probability of find it is high

Concepts to consider to compare performance w.r.t. classic

- Comparison made by consider total number of queries to a fixed oracle
- Efficiency of the circuit at most O(Poly(log(N))) two and one qubit gates
- Considered efficient if a quadratic speedup is possible wrt classical best search
- A speedup is possible w.r.t. some class of graphs (hypercube, toroid...)

Limitations

This method is limited to certain type of graphs:

- undirected graph
- weighted graphs

To overcome this limitations we need another model: Szegedy QW

Markov Chain

A Markov chain is

- Stochastic process
- sequence of random variables
- $P(X_n|X_{n-1}, X_{n-2}, ..., X_{n-N}) = P(X_n|X_{n-1})$
- if time-independent can be represented by a Transition Matrix P
- we can represent our graph with this

Graph representation

We can represent our graph with the Adjacency matrix

$$A_{i,j} = \begin{cases} 1 \, if (v_i, v_j) \in E \\ 0 \, otherwise \end{cases} \tag{7}$$

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Graph representation

Then we can construct the Transition matrix using as probability this equation:

$$P_{i,j} = \frac{A_{i,j}}{indeg(j)} \tag{8}$$

And we obtain:

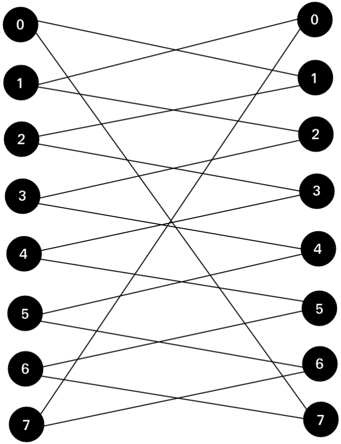
$$P = \frac{1}{2}\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Szegedy Quantum Walk

We need to define the components like in the Coined case:

Components

- mathematical space
- state vector
- swap operator
- Reflection operator



Szegedy Quantum Operator

The **mathematical space** for this operator is:

$$\mathcal{H} = \mathcal{H}_2^N \otimes \mathcal{H}_2^2 \quad (9)$$

Where N is the number of nodes, thus $\mathcal{H}$ has dimension N^2 Here the state vector is composed of two vector R_1 and R_2

State vector & Swap Operator

The **state vector** that represent the system is:

$$|\psi\rangle = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} a_{i,j} |i, j\rangle \tag{10}$$

The **Swap operator** S given by:

$$\mathcal{S} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |i, j\rangle \langle j, i| \tag{11}$$

Reflection Operator

Then we define the projector states of the Markov chain:

$$|\psi_i\rangle = |i\rangle \otimes \sum_{j=0}^{N-1} \sqrt{P_{j+1,i+1}}\,|j\rangle \equiv |i\rangle \otimes |\phi_i\rangle \tag{12}$$

where $|\phi_i\rangle$ is the square-root of the i-th column of the transition matrix P. The projector operator Then is given by:

$$\Pi = \sum_{i=0}^{N-1} |\psi i\rangle \langle \psi i| \tag{13}$$

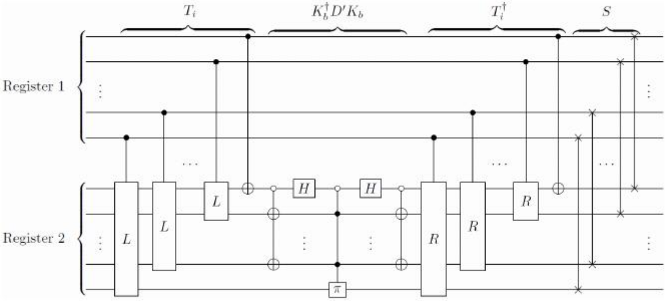
with the associated **Reflection Operator**:

$$\mathcal{R} = 2\Pi - I \tag{14}$$

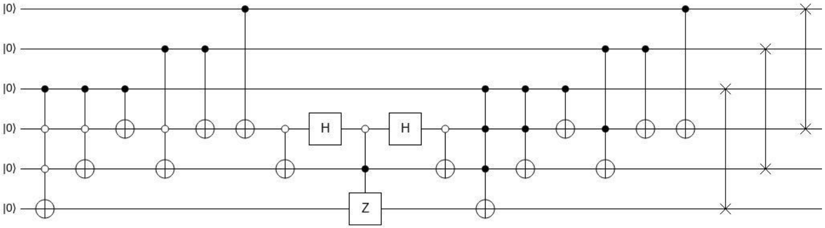
Szegedy QW Operator

Finally the one-step Szegedy QW operator is given by:

$$U_{walk} = S(I - 2\Pi) = SR \tag{15}$$



Circuit for the specific Szegedy QW on a cycle graph



Here the Quirk Simulation

Applications

Methods presented can be used and adapted to different areas, some example that i found are:

- NP hard solved with randomized algorithms
- Quantum page rank
- Hybrid linear system solver

The End

Thank you for your attention!