A quantum walk-based scheme for distributed searching on arbitrary graphs

Mathieu Roget, Giuseppe Di Molfetta

Aix-Marseille Université, Laboratoire d'informatique et systèmes

Introduction

Quantum Walk and Spatial Search

Quantum Walk (DTQW)

Register: linear combination of $|\text{position}\rangle \otimes |\text{coin}\rangle$.

Elementary step: one step is made of two operations: tossing the coin and scattering.

Scattering changes the position depending of the coin. Tossing the coin applies a unitary operator on the coin only (usually a 2x2 matrix).

Quantum Walk and Spatial Search

Quantum Walk (DTQW)

Register: linear combination of $|\text{position}\rangle \otimes |\text{coin}\rangle$.

Elementary step: one step is made of two operations: tossing the coin and scattering.

Scattering changes the position depending of the coin. Tossing the coin applies a unitary operator on the coin only (usually a 2x2 matrix).

Spatial search

A spatial search is a quantum walk to which we add an oracle. Every step first applies an oracle on the coin only for the marked position, then the coin, then the scattering.
The walker position will concentrate on the marked position over time.
The search complexity depends of the topology.

Goals and Motivations

- The definition of the QW usually depends of the topology. We would like a definition that works for any graph G.

Goals and Motivations

- The definition of the QW usually depends of the topology. We would like a definition that works for any graph G.
- Having a coin state of dimension 2 is convenient and we want to keep that.

Goals and Motivations

- The definition of the QW usually depends of the topology. We would like a definition that works for any graph G.
- Having a coin state of dimension 2 is convenient and we want to keep that.
- We want this quantum walk to be implemented with a Quantum Cellular Automata (QCA) to use their locality and translation invariance in a distributed framework.

Goals and Motivations

- The definition of the QW usually depends of the topology. We would like a definition that works for any graph G.
- Having a coin state of dimension 2 is convenient and we want to keep that.
- We want this quantum walk to be implemented with a Quantum Cellular Automata (QCA) to use their locality and translation invariance in a distributed framework.

Motivations : Distributed quantum computing

- Searching on arbitrary graphs without having to define a new model for every new topology.

Goals and Motivations

- The definition of the QW usually depends of the topology. We would like a definition that works for any graph G.
- Having a coin state of dimension 2 is convenient and we want to keep that.
- We want this quantum walk to be implemented with a Quantum Cellular Automata (QCA) to use their locality and translation invariance in a distributed framework.

Motivations : Distributed quantum computing

- Searching on arbitrary graphs without having to define a new model for every new topology.
- The locality of the quantum walk is well suite to tackle anonymous distributed problems.

Mathematical model

Quantum Walk

The Graph

$G = (V, E)$ an undirected connected graph with N vertices and M edges.

$d(u)$ the degree of node $u \in V$.

Polarity : Every edge has a polarity with a + side and a − side.

Quantum Walk

The Graph

$G = (V, E)$ an undirected connected graph with N vertices and M edges.

$d(u)$ the degree of node $u \in V$.

Polarity : Every edge has a polarity with a + side and a − side.

- The quantum walker moves onto the edges of G.

Quantum Walk

The Graph

$G = (V, E)$ an undirected connected graph with N vertices and M edges.

$d(u)$ the degree of node $u \in V$.

Polarity : Every edge has a polarity with a + side and a − side.

- The quantum walker moves onto the edges of G.
- The coin is of dimension 2.

Quantum Walk

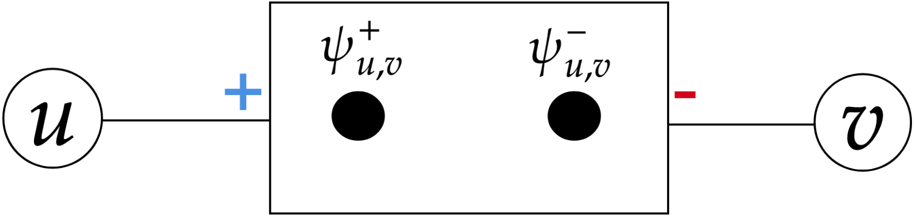
The Graph

$G = (V, E)$ an undirected connected graph with N vertices and M edges.

$d(u)$ the degree of node $u \in V$.

Polarity : Every edge has a polarity with a + side and a − side.

- The quantum walker moves onto the edges of G.
- The coin is of dimension 2.



Operations

Coin

The coin is a 2×2 unitary matrix C applied on every edge. For searching we use $C = X$.

Scattering

For the scattering we use the Grover diffusion operator locally around every node u

$$D_n = \begin{pmatrix} \frac{2}{n} - 1 & \frac{2}{n} & \cdots & \frac{2}{n} \\ \frac{2}{n} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{2}{n} \\ \frac{2}{n} & \cdots & \frac{2}{n} & \frac{2}{n} - 1 \end{pmatrix}.$$

Example

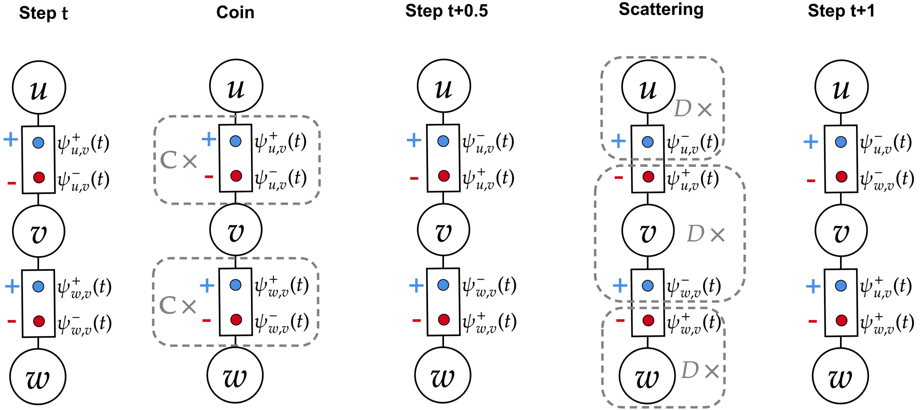


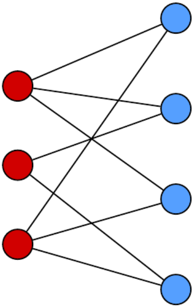
Figure: Example of a walk on a path of size 3.

Searching nodes

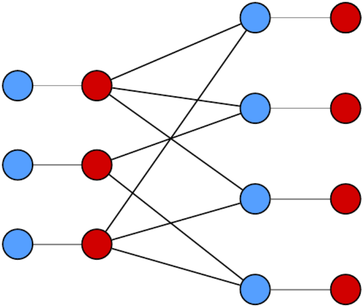
To search nodes instead of edges, we starify the graph and search for the virtual edge associated to the marked node.

Searching nodes

To search nodes instead of edges, we starify the graph and search for the virtual edge associated to the marked node.



(a) Initial Graph G.



(b) Starified graph $\tilde{G}$.

Figure: Transformation from the initial graph to the starified one. The starified graph is the initial graph to which we added one virtual node per real node. Every virtual node is connected to (and only to) the associated real node with a virtual edge.

Complexities

2D-Grid	Hypercube	Complete Graph	Scale-Free Graph
$O\left(\sqrt{M \log^\star M}\right)$	$O\left(\sqrt{M}\right)$	$O\left(\sqrt{M}\right)$	$O\left(\sqrt{M}\right)$
$O\left(\sqrt{N \log^\star N}\right)$	$O\left(\sqrt{N \ln N}\right)$	$O(N)$	$O\left(\sqrt{N+M}\right)$

Table: Complexity of the QWSearch for several graphs in function of M and N. Results are analytical for the complete graphs and numerical for the others. The classical complexity of a BFS algorithm is $O(M)$

Distributed implementation

Registers

Edge register

Two qubits per edge. This register is always a linear combination of the states $|\delta_k^{2M}\rangle$, where $\delta_k^n = \underbrace{0\ldots0}_{k-1 \text{ times}}\ 1\ \underbrace{0\ldots0}_{n-k \text{ times}}$. The starting state is the W state $\frac{1}{\sqrt{2M}}\sum_k |\delta_k^{2M}\rangle$. Measuring this register give us the position of the walker.

Registers

Edge register

Two qubits per edge. This register is always a linear combination of the states $|\delta_k^{2M}\rangle$, where $\delta_k^n = \underbrace{0\ldots0}_{k-1 \text{ times}}\ 1\ \underbrace{0\ldots0}_{n-k \text{ times}}$. The starting state is the W state $\frac{1}{\sqrt{2M}}\sum_k |\delta_k^{2M}\rangle$. Measuring this register give us the position of the walker.

Node register

We need $1+\log d$ qubits per node (d is the degree). These ancillary qubits starts with state $|0\rangle$ and are only useful to perform the scattering.

Example

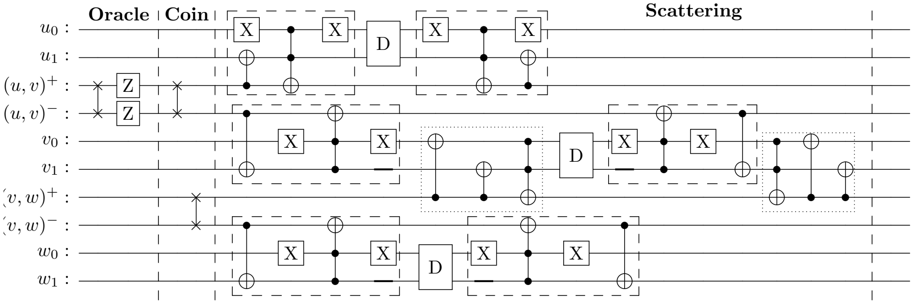


Figure: Circuit of one step of the quantum walk for the path graph $u - v - w$. Dashed lines signal Tr_1 circuit and its inverse while dotted lines Tr_2. The circuit applies successively the oracle on (u, v), the coin, Tr, D, Tr^{-1}.

Conclusion

Conclusion

- We have defined a new model of QW to search nodes and edges in graphs.
- The coin in our model is always of dimension 2.
- We have a quadratic speed-up when searching in several well known graphs including random scale-free graphs.
- We present a distributed scheme of this model using QCA.

Further works / Things to improve

- Complexity results are mostly numeric.
- The impact of changing the coin needs to be studied.
- Solving a distributed problem using this model.
- Looking at what happens when several nodes are marked.

Aknowledgement

This work is supported by the PEPR integrated project EPiQ ANR-22-PETQ-0007, by the ANR JCJC DisQC ANR-22-CE47-0002-01 founded from the French National Research Agency and with the support of the french government under the France 2030 investment plan, as part of the Initiative d'Excellence d'Aix-Marseille Université - A*MIDEX AMX-21-RID-011.

https://arxiv.org/abs/2310.10451