Reconfiguring staggered quantum walks with ZX*

Bruno Jardim, Jaime Santos, and Luis S. Barbosa

HASLab INESC TEC & Universidade do Minho,
Campus de Gualtar, 4710-057 Braga, Portugal

Abstract. The staggered model is a recent, very general variant of discrete-time quantum walks which, avoiding the use of a coin to direct the walker evolution, explores the underlying graph structure to build an evolution operator based on local unitaries induced by adjacent vertices. Optimising their implementation to increase resilience to decoherence phenomena motivates their analysis with the ZX-calculus. The whole optimisation can be seen as a graph reconfiguration process along which the original circuit is rewrote, significantly reducing the number of (expensive) gates used. The exercise identified an underlying pattern leading to an alternative, potentially more efficient evolution operator.

1 Introduction

Thought of as the quantum counterpart to classical random walks, quantum walks [VA12] provide an interesting technique in algorithmic design, with applications in unstructured search, graph algorithmics and communication protocols.

Differently from the classical case, where the walker's next move follows the result of some sort of random choice, in a quantum setting evolution typically proceeds in an equally weighed superposition of possible moves through the iteration of a unitary operator, without resorting to intermediate measurements. This results in a very rich dynamics, in which the design of the evolution operator, and even seemingly innocent differences in its phase and in the initial state, determine complex 'walking patterns' which differ greatly both from each other and from the classical setting.

The relevance of quantum walks as a tool for algorithmic design justifies both a better understanding of their behaviour and the optimisation of their implementation, namely to increase resilience to decoherence phenomena. This paper resorts to the ZX-calculus [CD08,vdW20,CHKW22] for such a purpose.

Optimisation of quantum circuits can be seen as a *reconfiguration* process. Indeed the interpretation of such circuits as ZX-diagrams provides a flexible description of quantum computations graphically. Then, the rules of the ZX-calculus guide through a simplification strategy which corresponds to sequences of graph transformations. Finally, the reconfigured circuit is extracted from the transformed graph. The process is illustrated here in a closed setting. However, it

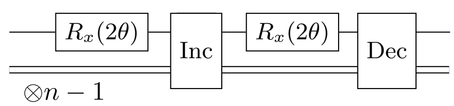
* This work was supported by FCT, the Portuguese Foundation for Science and Technology, within the project IBEX, with reference PTDC/CCI-COM/4280/2021.

extends smoothly to the dynamic case where algorithms reconfigure themselves as a result of the (classical) evaluation of measurement results. This is particularly relevant in the context of variational algorithms [CAB+21] currently used in quantum machine learning [DTB16].

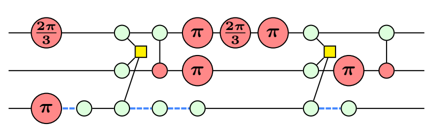
The exercise reported here focuses on a recent, very general variant of discrete-time quantum walks — the *staggered* model [PdOM17,PSFG16], briefly revisited in Appendix A — which, avoiding the use of a coin to direct the walker evolution, explores the underlying graph structure to build an evolution operator based on local unitaries induced by adjacent vertices. Section 2 discusses how its standard circuit implementation is translated and rewritten in ZX, supported by the *PyZX* tool [KvdW20a]. This process leads in Section 3 to the identification of a diagrammatic pattern providing an interesting approximation to, and in some cases more efficient version of, the underlying evolution operator.

2 Bringing ZX into the picture

A circuit implementation of the staggered model can be found in [San21]. For the example discussed in the Appendix, it yields



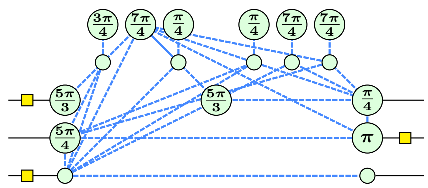
where $R_x(\theta) = e^{\frac{-i\theta X}{2}}$ and the Inc(rement) and Dec(rement) circuits have the usual implementation through generalised Toffoli gates. When the walker reaches the limit of the state space it cycles back. An implementation for a 3 qubit staggered quantum walk, and taking $\theta = \frac{\pi}{3}$, which maximizes[1] propagation, is represented in a ZX diagram as



using ZH-calculus H-box notation for a concise representation of Toffoli gates.

Advanced techniques, described in [KvdW20b] and directly implemented in *PyZX* [KvdW20a] as the `full_reduce` method, may reduce the circuit T-count in about 50% [KvdW20b]. Although this is not the case for our small example, when we start applying such simplifications to staggered models with larger amounts of steps the T-count reduction can reach approximately 60-70%. Back to the example, this simplification yields

[1] For this specific graph $\theta = \frac{\pi}{3}$ maximizes propagation, however $\frac{\pi}{2}$ is the optimal parameter for a complete graph.



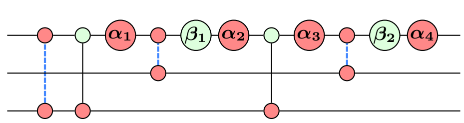
This diagram no longer resembles a circuit, making a comparison with the original one difficult. The circuit extracted [dBKvdW22] by *PyZX* has more gates than the one obtained from simple optimisations one can perform, although the T-count is indeed smaller. In fact, the `full_reduce` method introduces several additional Hadamard gates. The subsequent circuit extraction 'preserves' the nature of the *graph-like* ZX-diagram. Following the extraction with a small set of simplifications, basically resorting to fusion rewrite rules followed by a color-change, we get a much smaller circuit. This fully-simplified circuit now surpasses the original one in both the total amount of gates and T-count.Although this reduction is not outstanding in this example, it becomes most relevant when the number of steps in an example increases. The following tables show, respectively, the total number of gates and the T-count value induced by the different optimisation procedures for the same 3-qubit implementation.

	Number of steps in the staggered quantum walk:			
Optimizations used:	1	2	4	8
None	39	77	153	305
Full-reduce + fusion/id/to_rg	37	47	72	118

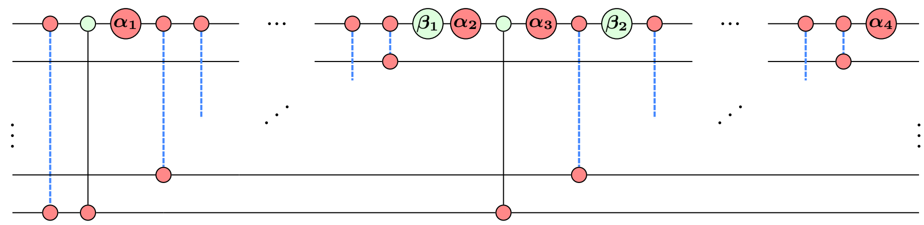
	Number of steps in the staggered quantum walk:			
Optimizations used:	1	2	4	8
None	16	32	64	128
Full-reduce + fusion/id/to_rg	10	16	28	52

3 An alternative evolution operator

When analysing the ZX diagram for a long staggered quantum walk (i.e. with more than 5 steps) a pattern starts to emerge, repeating itself as many times as the number of steps considered. Depicted in ZX below, it seems able to approximate, both the increment and decrement layers of the evolution operator.



where $\alpha_n = \pm\frac{\pi}{4}$ and $\beta_n = \frac{2\pi}{3} + m\pi$, with $m = 0$ or $m = 1$. There is also a slight variation of this operator, where a CNOT gate between the first and last qubit appears right after the β_1 Z-spider. This diagram does not fully capture the staggered model we started with, but, once suitably enveloped, it captures the exact same tensor as the original circuit. The set of gates to be placed as an envelope, in the beginning and the end of the diagram, does not exhibit a specific structure.This construction appeared when optimising the 3 qubit staggered quantum walk. However, it can be generalised for an n qubit implementation, yielding the following operator, in the form of a ZX-diagram:

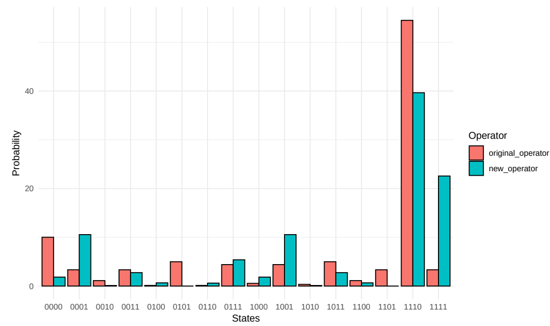


The rationale behind this operator is easy to explain: it creates a uniform distribution over a certain number of states, applies a rotation that makes some states more likely than others and then spreads these probabilities over the remaining states using CNOT gates. This also explains why the pattern only shows up in staggered quantum walks over a certain length. The classical evolution operator needs to be repeated a number of times to be able to spread the probability distributions over the whole state space. This is exactly what this version does on the first layer.

In any case, it has a number of advantages. First and foremost it reduces the total amount of gates needed to represent the evolution of the quantum walk. With the number of qubits increasing so does the cost of the increment and decrement layers, as a n qubit staggered quantum walk needs to implement MCX gates with $n-1$ controls. The alternative operator uses gates controlled by at most 1 qubit. Moreover, to go from an n-qubit to an $n+1$ qubit quantum walk, all that needs to be done is to add two more XCX-gates, one to each ladder of XCX-gates. In general, this makes the alternative operator much more efficient with respect to the total number of gates used, leading to lower depth and, therefore, potentially less error-prone circuits.

As mentioned above, just by itself this operator can approximate the evolution of a staggered quantum walk. Although the approximation is not perfect it can yield results which are quite similar to the ones obtained with the original implementation of the staggered model, as shown in the graph below, where the original operator is represented in red and the alternative in blue.

One particular advantage of this alternative evolution operator is that it can work quite well on a quantum processor with limited connectivity. This is due to the fact that all the qubits used in the staggered quantum walk only need to be strongly connected to the first qubit. However, a number of challenges remain, requiring further investigation. These concern the most suitable choice of parameters for α_n and β_n, as well as whether and how they depend on the number of qubits used in a particular staggered walk. Actually, when optimising the 4 qubit implementation of this circuit the resulting parameters did not seem to follow any regular pattern.



4 Conclusions and future work

This exercise showed that the original, 'intuitive' implementation of a staggered quantum walk can be heavily optimised with respect to both the total number of gates and the T-count value. It also lead to the identification of an alternative formulation of the evolution operator with a significant reduction in the number of gates involved and thus suitable for running on more limited quantum processing units. However, a number of issues, related to determining the suitable parametrisation scheme and better understanding the structure of the initial and final stages in the resulting circuit, still require further investigation. Similarly, it is not completely clear how the choice of the initial state influences how well the operator approximates the model evolution. Comparison of our results with other work on graph reconfiguration in ZX reported in recent references [DKPvdW20,UPR+23] is being carried out.

From another perspective, this exercise regards algorithmic optimisation in quantum programming as a *graph reconfiguration* process. This has a huge potential in the development of hybrid quantum-classical algorithms, which are the ones that can actually run in current quantum devices [Pre18]. They are essentially dynamic in the sense that, depending on a measurement carried over the quantum state, the quantum code running in the quantum device acting as a co-processor is transformed on-the-fly. The connection to suitable logic methods to reason about such transformations at a higher level of abstraction is a main direction for future work. The whole area of quantum machine learning and variational algorithms [DTB16,CAB+21] emerges as a main testbed for this research.

References

CAB+21. M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625?644, August 2021.

CD08. Bob Coecke and Ross Duncan. Interacting quantum observables. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings.*, volume 5126 of *Lecture Notes in Computer Science*, pages 298–310. Springer, 2008.

CHKW22. Bob Coecke, Dominic Horsman, Aleks Kissinger, and Quanlong Wang. Kindergarden quantum mechanics graduates *...or how I learned to stop gluing LEGO together and love the ZX-calculus. Theor. Comput. Sci.*, 897:1–22, 2022.

dBKvdW22. Niel de Beaudrap, Aleks Kissinger, and John van de Wetering. Circuit extraction for zx-diagrams can be #p-hard. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

DKPvdW20. Ross Duncan, Aleks Kissinger, Simon Perdrix, and John van de Wetering. Graph-theoretic simplification of quantum circuits with the zx-calculus. *Quantum*, 4:279, June 2020.

DTB16. Vedran Dunjko, Jacob M. Taylor, and Hans J. Briegel. Quantum-enhanced machine learning. *Physical Review Letters*, 117(13), September 2016.

KvdW20a. Aleks Kissinger and John van de Wetering. Pyzx: Large scale automated diagrammatic reasoning. *Electronic Proceedings in Theoretical Computer Science*, 318:229–241, May 2020.

KvdW20b. Aleks Kissinger and John van de Wetering. Reducing the number of non-clifford gates in quantum circuits. *Physical Review A*, 102(2), August 2020.

PdOM17. R. Portugal, M. C. de Oliveira, and J. K. Moqadam. Staggered quantum walks with hamiltonians. *Physical Review A*, 95(1), January 2017.

Pre18. John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018.

PSFG16. Renato Portugal, Raqueline A. M. Santos, Tharso D. Fernandes, and Demerson N. Gonçalves. The staggered quantum walk model. *Quantum Inf. Process.*, 15(1):85–101, 2016.

San21. Jaime Santos. Quantum random walks: simulations and physical realizations. MSc Thesis in Engineering Physics, DI, Universidade do Minho, 2021.

UPR$^+$23. Christian Ufrecht, Maniraman Periyasamy, Sebastian Rietsch, Daniel D. Scherer, Axel Plinge, and Christopher Mutschler. Cutting multi-control quantum gates with zx calculus. *Quantum*, 7:1147, October 2023.

VA12. Salvador Venegas-Andraca. Quantum walks: a comprehensive review. *Quantum Information Processing*, 11(5):1015–1106, July 2012.

vdW20. John van de Wetering. Zx-calculus for the working quantum computer scientist, 2020.

A Staggered quantum walks

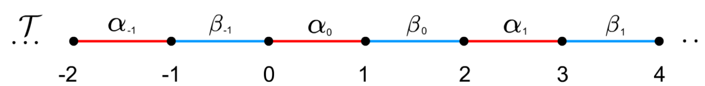
Staggered walks [PSFG16] explore partitions of graph cliques (subsets of vertices in which every two distinct vertices are adjacent) over the graph structure of the walking space. Each partition forms a tesselation whose elements do not overlap. The set of cliques in each tessellation must cover all vertices of the graph, and the set of tessellations $\{T_1, T_2, \ldots, T_k\}$ chosen must cover all the edges. Then a unit vector, typically encoding a uniform superposition, is associated to each clique so that the vector belongs to the subspace spanned by the corresponding vertices; i. e., $\left|u_j^k\right\rangle = \frac{1}{\sqrt{|\alpha_j^k|}} \sum_{l \in \alpha_j^k} |l\rangle$, where α_j^k is the j^{th} polygon in the k^{th} tessellation. This way each tessellation k gives rise to an operator $H_k = 2\sum_{j=1}^{p} \left|u_j^k\right\rangle \left\langle u_j^k\right| - I$. which propagates the probability amplitude locally, in each clique. The composition of all such operators defines the evolution operator, which, by solving the the time-independent Schrödinger equation, is equivalent to

$$U = e^{i\theta_k H_k} \ldots e^{i\theta_2 H_2} e^{i\theta_1 H_1}, \text{ where } \; e^{i\theta_k H_k} = \cos(\theta_k) I + i \sin(\theta_k) H_k$$

since $H_k^2 = I$, meaning that the Hamiltonian is a reflection operator that, when expanded in a Taylor series, generates a local operator.

As an elementary example consider a line where the following two tessellations (depicted in red and blue below) are defined

$$T_\alpha = \{\{2x, 2x+1\} \colon x \in \mathbb{Z}\} \; \text{ and } \; T_\beta = \{\{2x+1, 2x+2\} \colon x \in \mathbb{Z}\}.$$



Thus,

$$|\alpha_x\rangle = \frac{|2x\rangle + |2x+1\rangle}{\sqrt{2}} \quad \text{and} \quad |\beta_x\rangle = \frac{|2x+1\rangle + |2x+2\rangle}{\sqrt{2}},$$

yielding Hamiltonians

$$H_\alpha = 2\sum_{x=-\infty}^{+\infty} |\alpha_x\rangle \langle \alpha_x| - I \;\text{ and }\; H_\beta = 2\sum_{x=-\infty}^{+\infty} |\beta_x\rangle \langle \beta_x| - I.$$

Therefore, $U = e^{i\theta H_\beta} e^{i\theta H_\alpha}$ is the evolution operator. The probability distribution on a line after 50 steps, starting at $|+\rangle$, for different values of θ, is depicted below, noticing that the walker is more likely to be found further away from the origin as the angle increases.