An Algebraic Model for Simulation of Percolation Decoherence on Quantum-Walk-Based Search Algorithms

Gabriel Mauricio Oswald Vieira[1], Rui Aldé Lopes[1], Nelson Maculan[1], Franklin de Lima Marquezino[1,2]

[1]Federal University of Rio de Janeiro – Rio de Janeiro, RJ, Brazil

[2]University of Latvia – Riga, Latvia

{gvieira, rui, maculan, franklin}@cos.ufrj.br

Abstract. *In this paper, we propose a linear algebra based model for percolation noise simulation in quantum-walk-based searches to analyze the impact on the success probability of the search. The model was built to allow an open and simultaneous simulation of any type of noise that represents an edge-break, which we show to be a well-behaved family of permutation matrices that maintain a particular set theory relation when acting as some system evolution operators. The simulation evolves using only linear algebra operations, opening avenues for easier analyses of unexpected observed convergence behaviour. As a use case, we investigate the simulation of a lackadaisical quantum walk with percolation noise.*

1. Introduction

The laws of quantum mechanics state that the evolution of an isolated quantum system is deterministic. This work applies a probabilistic aspect in the evolution of the quantum system, which is the possibility that undesired, possibly exterior phenomena generate norm-preserving random noise and loss of information, i.e. decoherence, to analyze the impact on searching algorithms. Percolation, a particular kind of decoherence [Santos and Marquezino 2022], breaks edges of the graph where the quantum walk takes place.

The concept of quantum walks provides a versatile technique for the construction of quantum search algorithms [Santos and Marquezino 2022][Portugal 2013], which use quantum properties to find information stored within a particular data structure, a powerful tool for the simulation of complex physical systems [Abal et al. 2009]. In the beginning, this area was developed as a quantum version of the classic random walk, which requires the throwing of a coin to determine the direction of the next step.

Coin-based quantum walks have commonly been used in graphs without self-loops [Hoyer and Yu 2020]. The lackadaisical quantum walk proposed by Wong [Wong 2018] is an analogous model to the classical "lazy" random walk, which adds a self-loop to each vertex of the graph, in this case a two-dimensional lattice with cyclic boundary conditions, or a *torus*. The model presented in this paper can be used to simulate, on top of Wong's walk, the effect of decoherence on the search algorithm.

The phenomenon of decoherence has been widely studied in different kinds of quantum walks as introduced by [Abal et al. 2009], [Santos and Marquezino 2022], and it is the subject of Sec. 2. Our results are presented in Sec. 3. We present our conclusions in Sec. 4.

2. Algebraic Model for Percolation Decoherence

Quantum operations can be understood as matrices, and that allows the use of algebraic tools to re-imagine the algorithm. The lackadaisical quantum walk [Wong 2018] consists of 3 operations, Q, C and S, in each of the n iterations. The Q operation acts as the oracle in the algorithm, flipping the sign of the marked vertex, and C is the quantum coin operation, both of which have well understood matrix representations [Wong 2018]. The S operation is the shift used for the walk, which will be discussed further in the next paragraphs. This paper's model adds one or more decoherence operations to the sequence, represented by the letter B.

Similarly to Wong, all of these operations take place in the Hilbert space $\mathbb{C}^N \otimes \mathbb{C}^5$, using the N vertices of the graph and the 5 directions of the coin as a computational basis, and the algorithm begins with the same superposition $|\psi_0\rangle$, uniformly distributed over the vertices but not necessarily the coin directions. Then, after n steps, one could represent the reached state as

$$|\psi_n\rangle = \prod_{i=1}^{n} B_i S(I_n \otimes C)(Q \otimes I_5) |\psi_0\rangle \tag{1}$$

To explain B, it's important to remember that the S operation in this algorithm is a flip-flop shift, and can be understood as a permutation matrix that simply swaps the position of amplitudes in $|\psi\rangle$ according to the edges of the torus, giving the graph its structure. Hence, S permutates only specific pairs of positions in $|\psi\rangle$, in a controlled and predictable manner, and it follows that $S^{-1} = S = S^{\dagger}$.

What remains is to model the Broken Link as a matrix operation, that is, the probabilistic step in the algorithm where a percolation operation happens. If the Broken Link probability is 0, the operation should be equivalent to the identity matrix, as S must be unchanged. If the Broken Link probability is 1, the matrix should cancel out the flip-flop shift, once all links would be broken and the vertices would not permutate with each other at all, that is to say, $BS = I$. Since the S matrix is its own inverse, this would also imply that on the latter case we would have $B = S$.

To obtain these properties, B starts at each step identical to S. Then, for every edge in the graph, there is a probability of $(1 - p)$ that the two specific diagonal elements in the matrix encoding that connection, which are 0, become 1. Simultaneously, the off-diagonal, 1 elements in the same rows and columns become 0. This would effectively undo the shift in that specific edge, which is efficient in modeling the Broken Link property of the algorithm. A direct consequence of this model is that, for any B, it holds that $B^{-1} = B$.

One advantage of this approach is the possibility of concatenating several possible edge-breaking phenomena, using a different B matrix to represent each probabilistic step of losing some adjacency information in the structure of the graph. It is also important to note that the model allows the application of deterministic edge-breaking B_d simultaneously in the same step, as long as the matrices respect all the properties seen above. This is independent of any probabilistic decoherences B_p,

$$|\psi_n\rangle = \prod_{i=1}^{n} B_{d,i} B_{p,i} S(I_n \otimes C)(Q \otimes I_5) |\psi_0\rangle \, . \tag{2}$$

Finally, if two different matrices B_i and B_i' applied in the same step of the algorithm would break a common edge, the result would be the equivalent to keeping the edge intact, which might be desired or not, depending on the application of the model. As the quantity of edges in the graph is $2N$, there are only 2^{2N} different valid B matrices.

3. Application to the Lackadaisical Quantum Walk

The evolution operators in this study, when subjected to noise simulation, are obtained with an adaptation of the quantum walk presented by Wong [Wong 2018] and transcribed into the model presented above. To achieve that, a probabilistic instance of Broken Link noise is utilized, where the matrix permutations of type B_p are generated randomly, with a fixed Broken Link probability, so that

$$|\psi_n\rangle = \prod_{i=1}^{n} B_{p,i} S(I_n \otimes C)(Q \otimes I_5) |\psi_0\rangle . \tag{3}$$

Because S and B are both permutations, it is possible to represent them as set of disjoint pair of indices in $|\psi\rangle$ which will get swapped. Any swap that could possibly be in B must also be already in S. Thus, for any valid B, it follows that the permutations $B \subseteq S$, and it is also always possible to extract any valid sub-permutation $b \in S$ to compose a new B. Naturally, the model was constructed to be able to simulate scenarios more complex than a single instance of B_p for each step.

In the experiment applied in Wong [Wong 2018], the evolution of the success probability in finding the marked vertex in an environment without decoherence reaches optimal value when the weight of the self-loop probability is equal to d/N [Hoyer and Yu 2020][Wong 2018], where d is the degree of the regular graph and N is the order of the graph as shown in Figure 1.

In the results, it is possible to identify on Figure 1(a) that the success probability of the search converges to a value which appears to be close to the peak probability of the quantum walk without self-loop and without Broken Link. This remains true even for very high or very low Broken Link probabilities, and even in the worst cases it is noticeable within the first 10^7 steps.

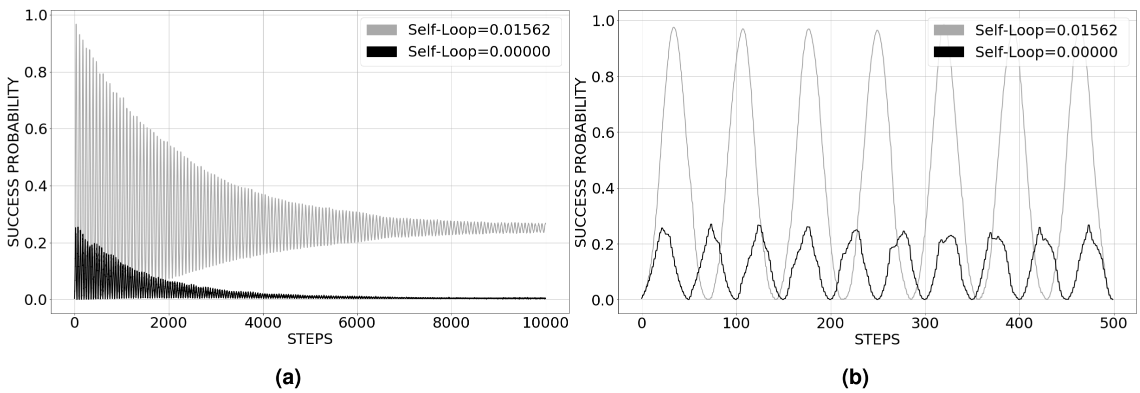

Figure 1. Sucess probability against time, for $d = 4.0$ and $N = 256$. (a) With broken link rate of 0.1%, averaged over 100 repetitions. (b) Without noise.

This phenomenon is not observed when the decoherence level is zero, that is to say, $B = I$, as in Figure 1(b). Figure 1(b) also hints at an interesting asymptotic behaviour, since as N grows the optimal self-loop weight, d/N, tends to zero, but setting the self-loop to be zero reduces the peak probability of success from close to 98% to close to 25%. The darker line in the graphs in Figure 1 represent a simulation with self-loop weight set to zero, and the lighter line represents the optimal weight of d/N.

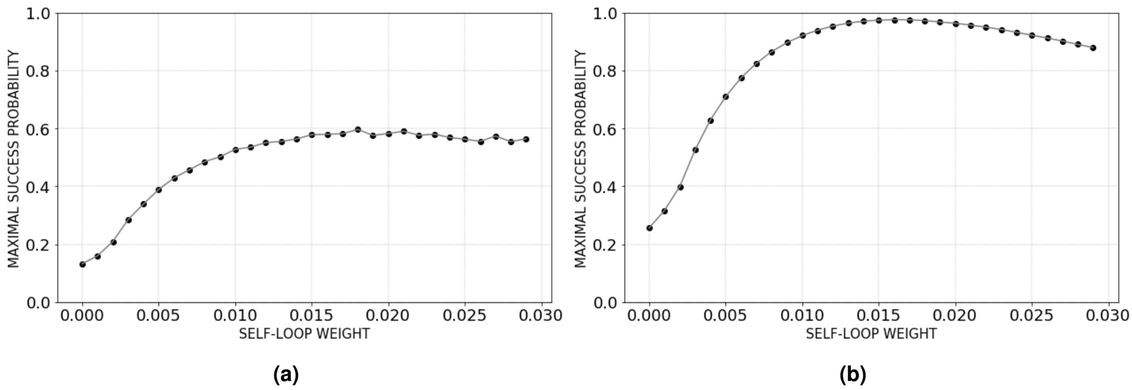

Figure 2. Success probability against weight of the self-loops. (a) With broken link rate of 3%, averaged over 100 repetitions. (b) Without noise.

The Figure 2(b) shows the evolution of the search success probability as a function of the variation in the self-loop weight in an environment without Broken Link.
Figure 2(a) shows the same evolution in a noisy environment. Broken Link = 3%. Each point in the graphs represents the maximum probability of a complete simulation of a search for a given self-loop weight.

4. Conclusion

The code for the simulator produced with the ideas presented in this text found consistent results with the expected and presented in Wong [Wong 2018], both with the presence of Broken Link and when there is only the lackadaisical quantum walk. The studied approach of using matricial algebra allows a greater flexibility and control in the creation and representation of different sources of decoherence in quantum algorithms. With it, it is possible to simulate any combination of edge-breaking decoherences, as long as we simply model them as B matrices and include them in the steps of the algorithm.

Furthermore, understanding the S and B operators as permutation matrices eases their comprehension and manipulation, and allows one to, for example, represent them as permutations, in which case we would have that the permutation B would be contained in the permutation S. This could allow in the future a possible approach using the tools and language of set theory to study these operations. Another unexplained and interesting aspect of the simulations is the success probability of the algorithm, when the amount of steps taken goes to infinity, which seems to converge.

Acknowledgments

R.A.L. thanks CAPES fellowship 88887.843699/2023-00. F.L.M. thanks CNPq grant 477018/2013-0, and Latvian Quantum Initiative, project 2.3.1.1.i.0/1/22/I/CFLA/001.

References

Abal, G., Donangelo, R., Marquezino, F. L., Oliveira, A. C., and Portugal, R. (2009). Decoherence in search algorithms. In *Proceedings of the XXIX Brazilian Computer Society Congress (CSBC)*, 36th Seminar on Software and Hardware (SEMISH), pages 293–306, Bento Conçalves, Brazil. Brazilian Computer Society.

Hoyer, P. and Yu, Z. (2020). Analysis of lackadaisical quantum walks. *Quantum Information and Computation*, 20(13 14):1138–1153.

Portugal, R. (2013). *Quantum Walks and Search Algorithms*. Springer.

Santos, R. A. M. and Marquezino, F. L. (2022). Decoherence on staggered quantum walks. *Physical Review A*, 105(3).

Wong, T. G. (2018). Faster search by lackadaisical quantum walk. *Quantum Information Processing*, 17(3).