Decoherence on staggered quantum walks

Raqueline A. M. Santos [*]
Center for Quantum Computer Science, Faculty of Computing, University of Latvia, Raina bulvaris 19, LV-1586 Riga, Latvia

Franklin de L. Marquezino [†]
Systems Engineering and Computer Science, Alberto Luiz Coimbra Institute for Graduate Studies and Research in Engineering,
Federal University of Rio de Janeiro, Avenida Horacio Macedo 2030, H319, 21941-972, Rio de Janeiro, Brazil
and Multidisciplinary Laboratory for Research in Computer Science, Federal University of Rio de Janeiro,
Rodovia Washington Luiz 19593, 25240-005, Duque de Caxias, Brazil

(Received 13 December 2021; accepted 7 March 2022; published 30 March 2022)

The decoherence phenomenon has been widely studied in different types of quantum walks. In this work we show how to model decoherence inspired by percolation on staggered quantum walks. Two models of unitary noise are described: breaking polygons and breaking vertices. The evolution operators subject to these noises are obtained, and the equivalence to the coined quantum walk model is presented. Further, we numerically analyze the effect of these decoherence models on the two-dimensional grid of 4-cliques. We examine how these perturbations affect the quantum walk based search algorithm in this graph and how expanding the tessellation intersection can make it more robust against decoherence.

DOI: 10.1103/PhysRevA.105.032452

I. INTRODUCTION

As the quantum counterparts of classical random walks, quantum walks have been defined in different ways in both discrete time and continuous time [1–3]. They are a powerful tool in the development of efficient quantum algorithms [4–6] and in the simulation of complex physical systems [7,8]. More recently, Portugal *et al.* [9] described a model of quantum walks on arbitrary graphs called the staggered quantum walk (SQW) model. This model was initially defined by covering the graph with tessellations. Each tessellation is a partition of the vertex set into polygons (or cliques). The relation of the SQW with other quantum walk models was studied in [9–13]. It has also been applied in the development of quantum algorithms [14–16] and in physical implementations [17].

It is known that when implementing quantum systems we can face decoherence problems. Quantum walk implementations are also affected by decoherence. It is important to understand, for example, when the classical behavior emerges and the quantum effect disappears. The quantum coherence is affected by the influence of random events which can be modeled in different ways. Refer to [18] for different methods of simulating decoherence in quantum walks. Decoherence inspired by percolation involves randomly removing or creating vertices or edges in the graph. This type of decoherence was analyzed in many research papers, using the discrete-time coined model [19–23], Szegedy's model [24], and the continuous-time quantum walk model [25–27].

Therefore, it is crucial to understand how decoherence affects the SQW model as well. In this paper, we analyze decoherence inspired by percolation on the SQW model. We describe two models of unitary noise: one that results from breaking vertices of the graph and another inherent in the SQW that results from breaking polygons. The latter is equivalent to breaking some edges of the graph, such that no additional tessellations are necessary. We describe how to obtain the evolution operators subject to these noises, and we show the equivalence to the coined quantum walk when the SQW can be obtained from the coined model. Moreover, we numerically study how decoherence affects the SQW on the two-dimensional grid of 4-cliques. We analyze the behavior of the quantum walk in terms of the displacement of the walker over time, and we examine how the perturbations affect the search for a marked vertex in this graph. Additionally, we observe how the search algorithm can be more robust against decoherence when we expand the tessellation intersection of the graph.

This paper is organized as follows. In Sec. II, we describe the SQW model. In Sec. III, we introduce the models of decoherence inspired by percolation and their equivalence to the coined model. In Sec. IV, we numerically analyze the effect of decoherence on the SQW on the two-dimensional grid of 4-cliques. Conclusions and further discussions are drawn in Sec. V.

II. THE STAGGERED QUANTUM WALK MODEL

The SQW model is defined by obtaining a *tessellation cover* of the graph. A *tessellation* $\mathcal{T}$ of a graph G is a partition of G into cliques. A *clique* of a graph G is a complete subgraph of G. A clique of size d is called a d-clique. Each clique in a tessellation is called a *polygon*. We say that an edge *belongs to* a tessellation if both end points of the edge belong to the same polygon. We denote by $\mathcal{E}(\mathcal{T})$ the set of edges belonging

[*] rsantos@lu.lv
[†] franklin@cos.ufrj.br

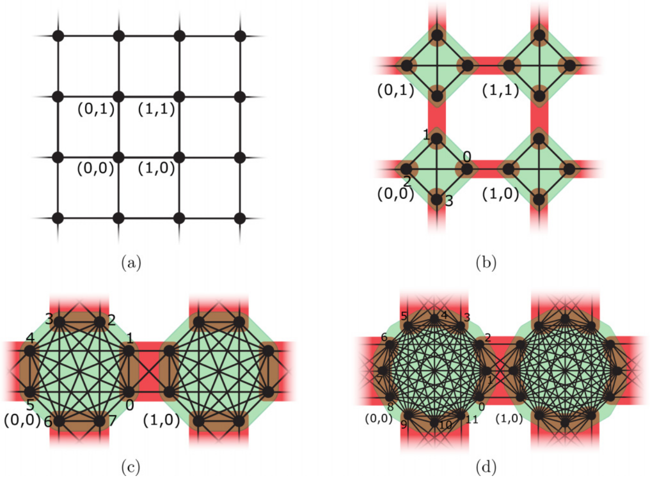


FIG. 1. Examples of SQWs on the two-dimensional grids of $4q$-cliques with two tessellations, green and red. (a) Two-dimensional grid. (b) Two-dimensional grid of 4-cliques. (c) Two-dimensional grid of 8-cliques. (d) Two-dimensional grid of 12-cliques. The SQW in (b) is the equivalent of the flip-flop coined quantum walk on the two-dimensional grid in (a).

to $\mathcal{T}$. Notice that a tessellation does not cover all the edges of a graph in general. A *tessellation cover* is a set $\{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$ of tessellations of a graph G such that the union $\bigcup_{i=1}^{k} \mathcal{E}(\mathcal{T}_i)$ is the edge set of G. The minimum number of tessellations required to cover a graph G is the *tessellation number*, denoted by $T(G)$. If a graph G is such that $T(G) \leqslant t$, for a fixed integer t, we say that G is t-tessellable.

Let us define a SQW on a connected simple graph G. Suppose G is ℓ-tessellable and the set $\{\mathcal{T}_1, \ldots, \mathcal{T}_\ell\}$ is a tessellation cover of G. Let $V(G)$ denote the set of vertices of G and $|V(G)| = N$. The N-dimensional Hilbert space $\mathcal{H}^N$ of the SQW is spanned by $\{|v\rangle : v \in V(G)\}$. Each basis state $|v\rangle$ is associated with a vertex v of the graph. Suppose each tessellation $\mathcal{T}_k$ has $|\mathcal{T}_k|$ polygons denoted by P_i^k. Each polygon is associated with a unit vector

$$\left|P_i^k\right\rangle = \sum_{v \in V(P_i^k)} \alpha_{\text{vik}} \, |v\rangle\,, \tag{1}$$

with

$$\sum_{v \in V(P_i^k)} |\alpha_{\text{vik}}|^2 = 1. \tag{2}$$

Each tessellation $\mathcal{T}_k$ is associated with a Hermitian operator

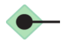
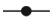
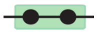
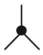
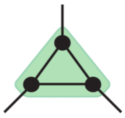
$$U_{\mathcal{T}_k} = 2 \sum_{j=1}^{|\mathcal{T}_k|} \left|P_j^k\right\rangle\!\left\langle P_j^k\right| - I. \tag{3}$$

The evolution operator of the SQW is given by

$$U = U_{\mathcal{T}_\ell} \cdots U_{\mathcal{T}_2} U_{\mathcal{T}_1}. \tag{4}$$

For example, let us consider a SQW on two-dimensional grids of $4q$-cliques with periodic boundary conditions, as depicted in Fig. 1.These graphs will be used throughout the paper. Figures 1(b)–1(d) depict the cases where $q = 1, 2, 3$, respectively.

TABLE I. Converting a vertex v of degree $d(v)$ into a $d(v)$-clique with the associated green polygon belonging to the coin tessellation $\mathcal{T}_C$.

$d(v)$	v	$d(v)$-clique
1		
2		
3		

The graph is composed of n^2 $4q$-cliques linked by $2n^2$ $2q$-cliques with a toruslike topology. The Hilbert space associated with the graph has dimension $4qn^2$. The vectors associated with the green polygons are

$$|\alpha_{xy}\rangle = \frac{1}{2\sqrt{q}} \sum_{k=0}^{4q-1} |x, y, k\rangle\,, \tag{5}$$

and the vectors associated with the red polygons are

$$\left|\beta_{xy}^{(0)}\right\rangle = \frac{1}{\sqrt{2q}} \sum_{k=0}^{q-1} (|x, y, k\rangle + |x+1, y, 2q+k\rangle), \tag{6}$$

$$\left|\beta_{xy}^{(1)}\right\rangle = \frac{1}{\sqrt{2q}} \sum_{k=0}^{q-1} (|x, y, q+k\rangle + |x, y+1, 3q+k\rangle) \tag{7}$$

for $0 \leqslant x, y \leqslant n-1$. The arithmetic with the labels of $|x, y\rangle$ is performed modulo n.

The evolution operator is $U = U_1 U_0$, where

$$U_0 = 2 \sum_{x,y=0}^{n-1} |\alpha_{xy}\rangle\,\langle\alpha_{xy}| - I \tag{8}$$

and

$$U_1 = 2 \sum_{x,y=0}^{n-1} \left|\beta_{xy}^{(0)}\right\rangle\!\left\langle\beta_{xy}^{(0)}\right| + \left|\beta_{xy}^{(1)}\right\rangle\!\left\langle\beta_{xy}^{(1)}\right| - I. \tag{9}$$

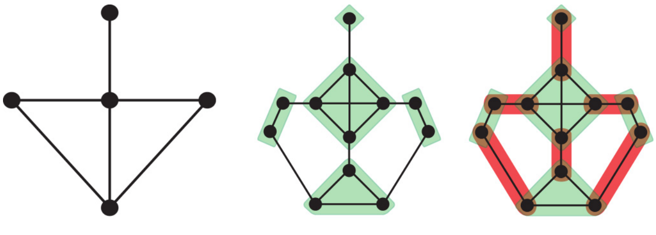


FIG. 2. (a) Graph G. (b) Graph G' with the coin tessellation in green. (c) Graph G' with the coin tessellation in green and shift tessellation in red. Notice that graph G' in (b) is obtained from G in (a) by substituting each vertex $v \in V(G)$ with a $d(v)$-clique, which belongs to a green polygon of the coin tessellation $\mathcal{T}_C$. In (c) the shift tessellation $\mathcal{T}_S$ in red is obtained for each pair of connected vertices from different green polygons.

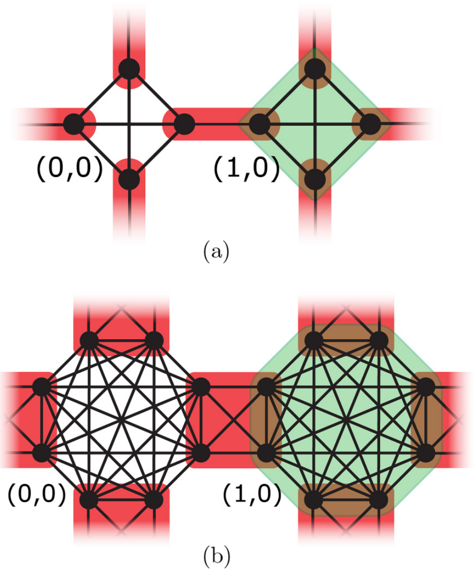


FIG. 3. Search for the $4q$-clique in position (0,0). The green polygon containing the clique in position (0,0) is removed, and we have a SQW with a partial tessellation. (a) $q = 1$. (b) $q = 2$.

A. Equivalence to the coined model

Consider a flip-flop coined quantum walk (FCQW) on a graph G. The Hilbert space of the walk is $\mathcal{H}^{2|E|}$, where $|E|$ is the number of edges of G. The basis state $|u, v\rangle$ represents the state of the walker at vertex u pointing in the direction of vertex v. This walk is driven by the unitary operator $U = SC$, where C is the coin operator that modifies the internal coin state of the walker and S is the flip-flop shift which is responsible for moving the walker between adjacent vertices, $S|u, v\rangle = |v, u\rangle$. Reference [28] gives a detailed description of the coined quantum walk model. It is possible to obtain the equivalent SQW of the FCQW, as we briefly describe here. For more details, see [10].

The SQW takes place in a space with the dimension equal to the number of vertices of the graph, whereas the FCQW uses a bigger dimension space. In order to obtain the SQW equivalent of an FCQW, first, we should create a new graph G', so the dimensions will be equivalent. In this new graph, we convert each vertex $v \in V(G)$ into a $d(v)$-clique, where $d(v)$ is the degree of vertex v.

Each created clique will belong to the coin tessellation $\mathcal{T}_C$, as we can see in Table I. The 2-clique formed by the connection between vertices of two different polygons in $\mathcal{T}_C$ will belong to the shift tessellation $\mathcal{T}_S$. See Fig. 2 for an example. The coin tessellation is represented in green, and the shift tessellation is shown in red. Notice also that Fig. 1(b) is the SQW equivalent of the FCQW on the two-dimensional grid [Fig. 1(a)].

We can obtain exactly the shift and coin operators from the operators generated by each tessellation, that is, $U_{\mathcal{T}_C} \equiv C$ and $U_{\mathcal{T}_S} \equiv S$, as long as the coin operator is an orthogonal reflection. This condition is required in order to obtain the equivalent SQW. See [10] for a complete description of what an orthogonal reflection is and the whole equivalence process. The most used Grover and Hadamard coin operators are orthogonal reflections.

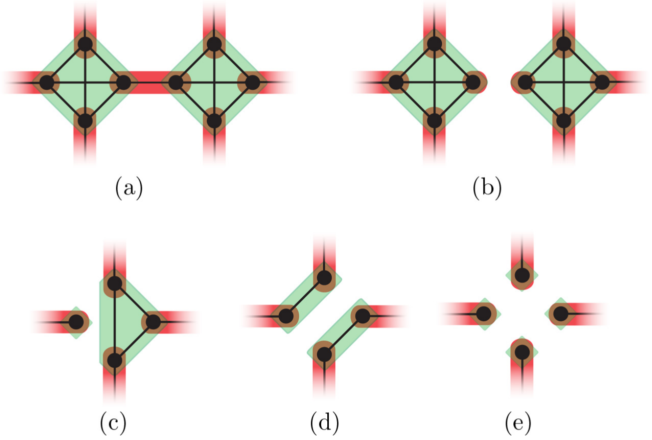


FIG. 4. Example of breaking polygons in the SQW model. (a) A two-tessellable graph with green and red tessellations. (b) Breaking a red polygon. (c) Breaking a green polygon into a 1-clique polygon and a 3-clique polygon. (d) Breaking a green polygon into two 2-clique polygons. (e) Breaking a green polygon into four 1-clique polygons.

B. Search with SQWs

The implementation of spatial search on SQWs can be done by using partial tessellations. This is simply done by removing polygons from the tessellation. The vertices in the missing polygons will be the marked ones [9].

Consider the example for the two-dimensional grid of $4q$-cliques presented earlier. Suppose that the $4q$-clique in position (0,0) is the marked one, as depicted in Fig. 3 for $q = 1$ and $q = 2$. We remove the polygon which induces the state

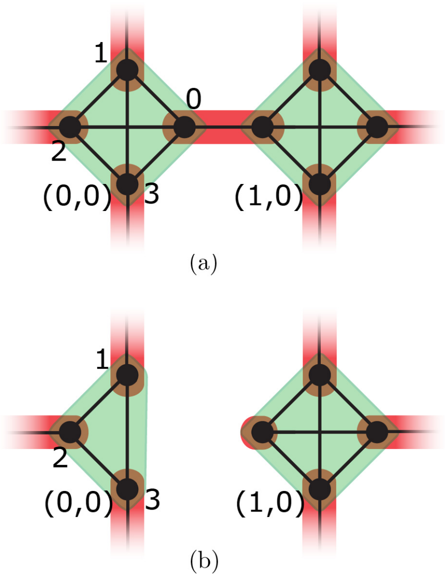


FIG. 5. Example of breaking a vertex in the SQW model. (a) A two-tessellable graph with green and red tessellations. (b) Removing vertex 0 in position (0,0).

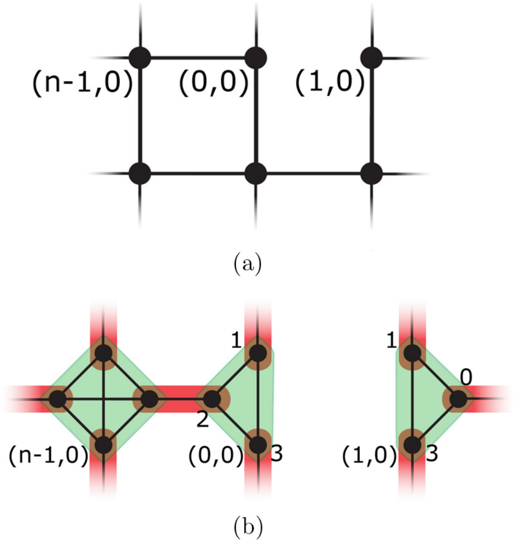


FIG. 6. Example of breaking an edge in the coined version and its equivalent in the staggered model. (a) The two-dimensional grid with a broken edge. (b) Two vertices removed in the SQW equivalent of the FCQW version.

$|\alpha_{00}\rangle$ in order to obtain the search operator $U' = U_1 U_0'$, where

$$U_0' = 2 \sum_{\substack{x,y=0 \\ (x,y)\neq(0,0)}}^{n-1} |\alpha_{xy}\rangle \langle\alpha_{xy}| - I.$$

The initial state $|\psi(0)\rangle$ is the uniform superposition of all vertices of the graph, and the state at time t is given by $|\psi(t)\rangle = U'^{t} |\psi(0)\rangle$.

Portugal *et al.* [9] showed that searching for a marked vertex using FCQW with coin $-I$ on the marked vertex is equivalent to searching for a missing polygon in the equivalent SQW. This means that searching with SQW for a missing green polygon in the two-dimensional grid of 4-cliques [Fig. 3(a)] is equivalent to searching for a marked vertex in the two-dimensional grid using FCQW.

Recently, Santos [29] showed that by a process called *intersection expansion* (*reduction*) we can add (remove) vertices to some tessellation intersection of the SQW. The dynamics of the resulting SQW on the new graph will be equivalent to the dynamics on the original graph if some assumptions are made for the vertices in the intersection. The two-dimensional grids of $4q$-cliques can be obtained from each other by a process of tessellation intersection or reduction. Searching for a clique on the green polygons in any of these graphs is equivalent [29]. The number of steps of the algorithm is $O(\sqrt{N}\log N)$, and the success probability is $O(1/\log N)$, where $N = n^2$ and n is the width of the grid. The total cost of the algorithm after applying the amplitude amplification method is $O(\sqrt{N}\log N)$.

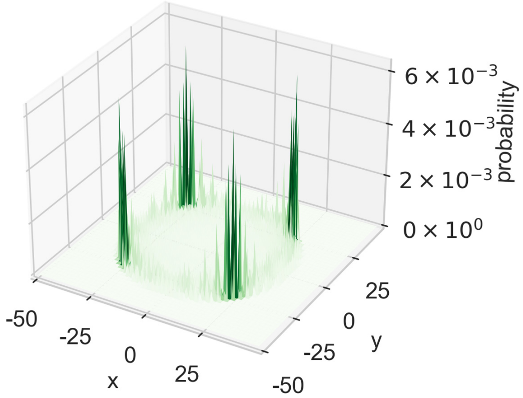


FIG. 7. Probability distribution of the two-dimensional grid for the SQW on the two-dimensional grid of 4-cliques with $n = 100$ at $t = 50$. The initial state is given by Eq. (23).

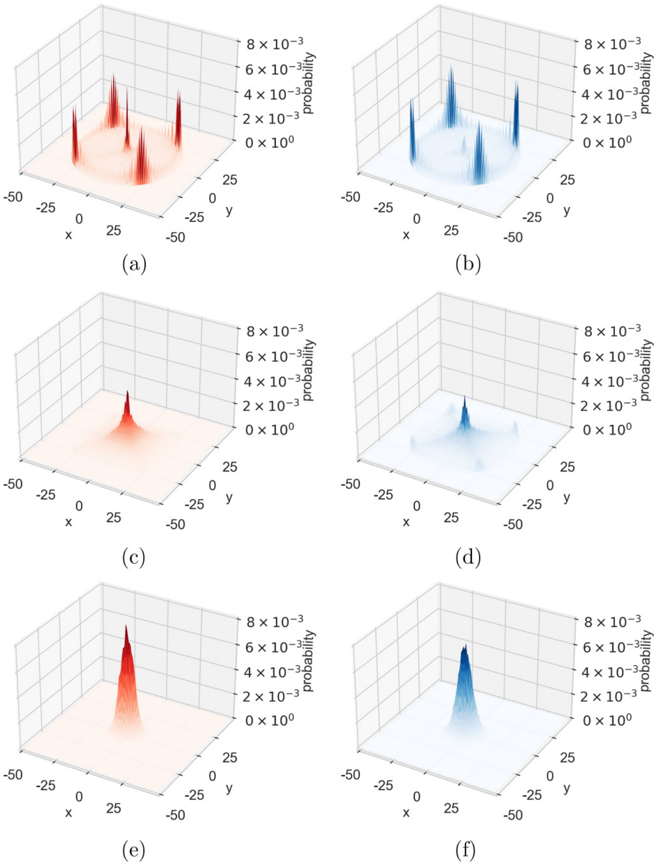


FIG. 8. Average of the probability distribution over 100 runs for the SQW on the two-dimensional grid of 4-cliques with $n = 100$. (a) Breaking vertices, $p = 0.001$. (b) Breaking polygons, $p = 0.001$. (c) Breaking vertices, $p = 0.01$. (d) Breaking polygons, $p = 0.01$. (e) Breaking vertices, $p = 0.1$. (f) Breaking polygons, $p = 0.1$.

III. DECOHERENCE INSPIRED BY PERCOLATION

Decoherence inspired by percolation allows removing vertices and/or edges in the graph. What happens when we break edges and/or vertices in the SQW model? Is it possible to “break” polygons as well? The staggered model is described by a graph tessellation cover in order to obtain the evolution operator. The tessellations and its polygons follow some strict properties, as we mentioned before. In order to model this kind of decoherence we should take care with the tessellation structure so that its properties are satisfied and the evolution continues to be unitary. In the following, we describe some

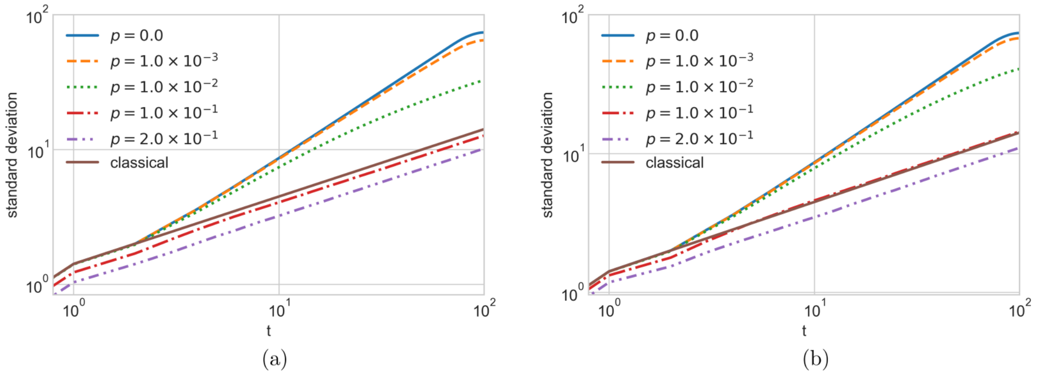


FIG. 9. Standard deviation for the SQW on the two-dimensional grid of 4-cliques with $n = 100$ considering the decoherence model of (a) breaking vertices and (b) breaking polygons in the cases with $p = \{0.0, 0.001, 0.01, 0.1, 0.2\}$. The solid (lower) brown curve depicts the classical case on the two-dimensional grid.

possible models of decoherence inspired by percolation affecting the SQWs. The models are valid for general graphs despite the tessellation number.

A. Breaking polygons

A polygon can be broken or divided into many polygons (as many as the number of vertices of the clique). Breaking a polygon is associated with breaking some edges of the clique, as we can see from the example in Fig. 4. A polygon in the green tessellation can be broken in different ways [see Figs. 4(c)–4(e)], whereas a red polygon can be broken only into two 1-clique polygons [see Fig. 4(b)]. That is because each red polygon is a 2-clique in this case.

Let us assume that polygon P_ℓ^j was broken into $m \in \{2, \ldots, |P_\ell^j|\}$ polygons. Each new polygon $P_{\ell i}^j (i = 1, \ldots, m)$ is a clique formed by a subset of vertices of the original polygon P_ℓ^j. The set $\{V(P_{\ell i}^j) : i = 1, \ldots, m\}$ is mutually disjoint, and

$$V\left(P_\ell^j\right) = \bigcup_{i=1}^{m} V\left(P_{\ell i}^j\right). \tag{10}$$

In the state associated with each new polygon $P_{\ell i}^j$, the probability flux that was going to the vertices in $P_\ell^j \backslash P_{\ell i}^j$ should be diverted into the vertices in $P_{\ell i}^j$ so that the state remains unitary, that is,

$$\left|P_{\ell i}^j\right\rangle = \sum_{v \in V(P_{\ell i}^j)} \frac{\alpha_{v\ell j}}{\beta_{\ell i j}} \left|v\right\rangle, \tag{11}$$

where

$$\beta_{\ell i j} = \sqrt{\sum_{v \in V(P_{\ell i}^j)} |\alpha_{v\ell j}|^2} \tag{12}$$

is the normalization factor. The reflection operator associated with tessellation $\mathcal{T}_j$ has now become

$$U_{\mathcal{T}_j} = 2\left(\sum_{\substack{k=1\\k\neq\ell}}^{|\mathcal{T}_j|} \left|P_k^j\right\rangle\left\langle P_k^j\right| + \sum_{i=1}^{m} \left|P_{\ell i}^j\right\rangle\left\langle P_{\ell i}^j\right|\right) - I. \tag{13}$$

The evolution operator of the SQW is given by (4).

Let us see an example for the two-dimensional grid of 4-cliques [Fig. 1(b), $q = 1$]. Assume that the polygon in position (0,0) is broken into four 1-clique polygons, as depicted in Fig. 4(e). That is, the polygon which is associated with the quantum state

$$|\alpha_{00}\rangle = \frac{1}{2}\sum_{k=0}^{3} |x, y, k\rangle \tag{14}$$

is broken into the polygons

$$|\alpha_{00v}\rangle = |v\rangle, \qquad v \in \{0, 1, 2, 3\}. \tag{15}$$

The reflection operator associated with the green tessellation is now described as

$$U_0 = 2\left(\sum_{\substack{x,y=0\\(x,y)\neq(0,0)}} |\alpha_{x,y}\rangle\,\langle\alpha_{x,y}| + \sum_{v=0}^{3} |\alpha_{00v}\rangle\,\langle\alpha_{00v}|\right) - I. \tag{16}$$

The operator associated with the red tessellation is unchanged.

Equivalence to the flip-flop coined QW model

Now consider we have the staggered model obtained from the coined model. In this case, there are two types of tessellations. One tessellation is associated with the coin action, and the other is associated with the shift action. For example, in Fig. 1(b), if we consider that the graph is obtained from a flip-flop coined QW on the two-dimensional grid, the green tessellation is associated with the coin operator, and the red tessellation is associated with the shift operator.

Remember that the polygons of the shift tessellation consist of 2-cliques. The only way to break a polygon in this case is to break it into two 1-clique polygons. Therefore, breaking a polygon of the shift tessellation will be equivalent to not applying the shift operator to the related edge in the FCQW version. We can think of it as removing an edge in the FCQW, but recall that the coin operator will still act as if there is an edge in this case.

Breaking polygons of the coin tessellation will be equivalent to modifying the coin in the associated vertex of the FCQW version. For example, if we break the polygon into

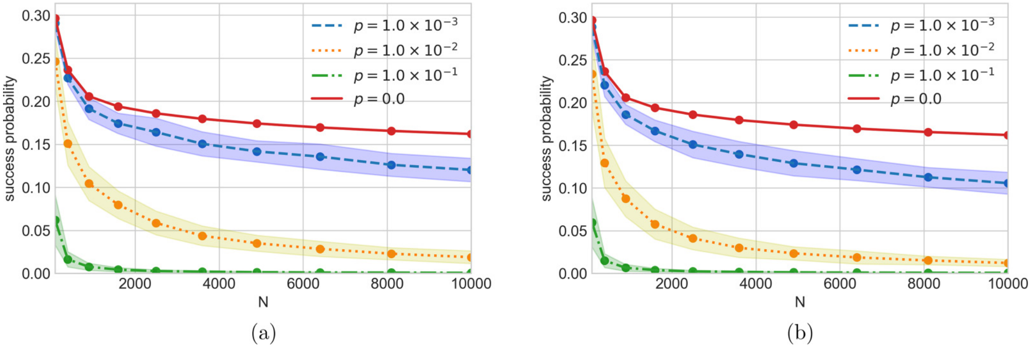


FIG. 10. The average success probability for $n = 10, 20, \dots, 100$ ($N = n^2$). The solid red curve is for $p = 0$, the dashed blue curve is for $p = 0.001$, the dotted orange curve is for $p = 0.01$, and the dot-dashed green curve is for $p = 0.1$. The light area around the curves shows the confidence bands. (a) Breaking vertices. (b) Breaking polygons into 1-clique polygons.

1-clique polygons, then it will be equivalent to not applying the coin operator in the related vertex of the FCQW version (or the coin is the identity).

B. Breaking vertices

Now suppose we remove a vertex from the graph. In this case, we have to make arrangements in all the tessellations (since a tessellation covers all vertices in the graph). We should remove the vertex from the polygons which contains it. Since each polygon is a clique, by removing one of its vertices it will continue to be a clique (not breaking any properties). Let us suppose vertex v is removed. Then, the probability flux that was going to the removed vertex should be diverted to the other vertices. For every polygon P_i^j that contains vertex v, the state associated with the polygon becomes

$$\left|P_i^j\right\rangle = \sum_{w \in V(P_i^j)\backslash\{v\}} \frac{\alpha_{wij}}{\beta_{ij}} \left|w\right\rangle, \tag{17}$$

where β_{ij} is the normalization factor so that the state can remain unitary, that is,

$$\beta_{ij} = \sqrt{\sum_{w \in V(P_i^j)\backslash\{v\}} |\alpha_{wij}|^2}. \tag{18}$$

For the example shown in Fig. 5(b), the state associated with the green polygon, which contains the removed vertex (0,0,0),

$$|\alpha_{00}\rangle = \frac{1}{2}\sum_{k=0}^{3} |0, 0, k\rangle \tag{19}$$

becomes

$$|\alpha_{00}\rangle = \frac{1}{\sqrt{3}}\sum_{k=1}^{3} |0, 0, k\rangle, \tag{20}$$

and the state associated with the red polygon, which contains the removed vertex (0,0,0),

$$\left|\beta_{00}^{(0)}\right\rangle = \frac{1}{\sqrt{2}}(|0, 0, 0\rangle + |1, 0, 2\rangle) \tag{21}$$

becomes

$$\left|\beta_{00}^{(0)}\right\rangle = |1, 0, 2\rangle. \tag{22}$$

The operators U_0 and U_1 associated with each tessellation are obtained as before, considering the new states given by (20) and (22).

Equivalence to the flip-flop coined QW model

Consider the staggered model obtained from the coined model. Removing a vertex in the SQW model affects the shift and coin tessellations. If the removed vertex belongs to a 1-clique polygon of the coin tessellation, then it will be equivalent to removing the associated vertex in the coined model. Otherwise, it will be equivalent to modifying the action of the coin and shift operators in the vertex of the FCQW associated with the clique which contains the removed vertex. Removing a vertex in the SQW model also affects the shift action on the neighbor vertex of the FCQW associated with its direction. We can think of it almost as removing an edge in the coined version. Only if we remove the clique which belongs to the polygon in the shift tessellation will it be equivalent to removing an edge in the coined version, as depicted in Fig. 6.

IV. NUMERICAL SIMULATIONS

Consider the SQW on the two-dimensional grid of 4-cliques [Fig. 1(b)]. This graph is the equivalent of the FCQW on the two-dimensional grid. Let the initial state of the SQW be

$$|\psi(0)\rangle = \frac{1}{2}\sum_{k=0}^{3} |0, 0, k\rangle, \tag{23}$$

which means we are starting at position (0,0). The probability of obtaining the position (x, y) at time t after measurement in the computational basis is given by

$$P_t(x, y) = \langle\psi(t)| \left(|x, y\rangle\langle x, y| \otimes I_4\right) |\psi(t)\rangle; \tag{24}$$

that is, we sum the probability at each vertex of the clique in position (x, y) from the state at time t.

From (24), we can obtain the probability distribution of the walk on the two-dimensional grid as depicted in Fig. 7 for $n = 100$, $t = 50$, and initial state (23). We can observe that the walk is symmetric and delocalized. The same behavior can be observed for the non-flip-flop coined QW with Grover and other coins [20].

Now we analyze what happens to the SQW when it is affected by the decoherence models described in Sec. III. Let p be the probability of breaking each polygon or vertex of the graph before step t of the quantum walk; that is, we consider a model of dynamic percolation. In the case of breaking polygons, we break them into 1-clique polygons. Figure 8 shows the behavior of the probability distribution for $p \in \{0.001, 0.01, 0.1\}$. The quantum behavior of delocalization is still quite present for $p = 0.001$ [Figs. 8(a) and 8(b)]. The classical behavior starts to develop as we increase the value of p. For the breaking-polygon case, it seems the classical behavior emerges a little bit slower than the breaking-vertex case. For $p = 0.1$ both plots [Figs. 8(e) and 8(f)] present a fully developed classical behavior.

The transition from the quantum to the classical behavior can be better observed with the plot of the standard deviation σ, as depicted in Fig. 9 for $n = 100$. Figure 9(a) shows the standard deviation for the breaking-vertex case, and Fig. 9(b) shows the case for the breaking-polygon case. The solid (upper) blue curve is the quantum case without decoherence ($p = 0$). Notice that there is an inclination of the curve when reaching $t = 100$, which happens because we have periodic bounded conditions of the grid. The solid (lower) brown curve shows the classical case. We can observe how the walk loses its quantum behavior when the value of p increases. A similar behavior can be seen for the coined QW with the Grover coin on the two-dimensional grid with broken links [20].

How decoherence affects searching

Suppose we are searching for a missing green polygon on the two-dimensional grid of 4-cliques. This is equivalent to searching for a marked vertex in the two-dimensional grid using the FCQW model, as mentioned before. We analyze the decoherence models described in Sec. III. Let p be the probability of breaking each polygon or vertex of the graph before step t of the quantum walk. In the case of breaking polygons, we break them into 1-clique polygons. We run the simulations for $p \in \{0.001, 0.01, 0.1\}$ until the maximum number of steps is achieved. According to some simulations, the maximum number of steps of $1.5\sqrt{N \log N}$ is enough to obtain the maximum peak of the success probability. We obtain the average over 100 runs for each value of p. The results are compared with the simulation without decoherence ($p = 0$).

Figure 10 shows the average probability of finding the marked vertex, that is, the success probability, for $n = 10, 20, \ldots, 100$. The case of breaking vertices is depicted in Fig. 10(a), and the case of breaking polygons is shown in Fig. 10(b). The solid red curve represents $p = 0$, the dashed blue curve shows $p = 0.001$, the dotted orange curve shows $p = 0.01$, and the dot-dashed green curve represents $p = 0.1$. The confidence bands for each simulated probability are depicted by the light area around each curve. As expected, the value of the success probability decreases as the value of p increases. All curves for $p \in \{0.001, 0.01, 0.1\}$ are below the curve for $p = 0$.

Figure 11 shows the average running time, i.e., the number of steps to reach the maximum of the average success probability divided by the square root of the average success probability, which is due to the complexity of the amplitude amplification algorithm [30]. The breaking-vertex and breaking-polygon cases are depicted in Figs. 11(a) and 11(b), respectively. The dashed blue curve ($p = 0.001$) is quite close to the case without decoherence. The running time increases as the value of p increases. For $p = 0.01$ the curve starts to have a higher inclination than the case for $p = 0$. The behavior for $p = 0.1$ is already quite erratic since for this value of probability the walk has already developed its classical behavior, as shown before.

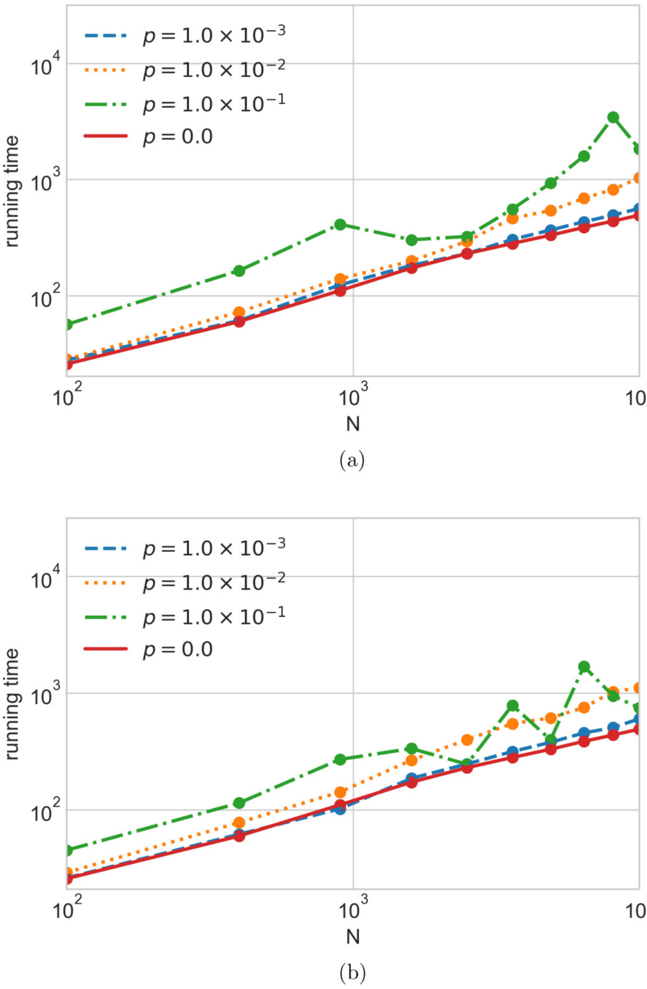


FIG. 11. The average running time for $n = 10, 20, \ldots, 100$ ($N = n^2$) in logarithmic scale. The solid red curve is for $p = 0$, the dashed blue curve is for $p = 0.001$, the dotted orange curve is for $p = 0.01$, and the dot-dashed green curve is for $p = 0.1$. (a) Breaking vertices. (b) Breaking polygons into 1-clique polygons.

Expanding the tessellation intersection

As mentioned before, searching for a clique on the green polygons in any of the two-dimensional grids of $4q$-cliques is equivalent [29]. Let us analyze how robust against decoherence the SQW can be as we increase the value of q. We analyze the decoherence model of breaking polygons. For this case, we consider that each polygon has probability p of being broken into two polygons: one with a 1-clique and the other containing a $(4q - 1)$-clique. The vertices in each of the

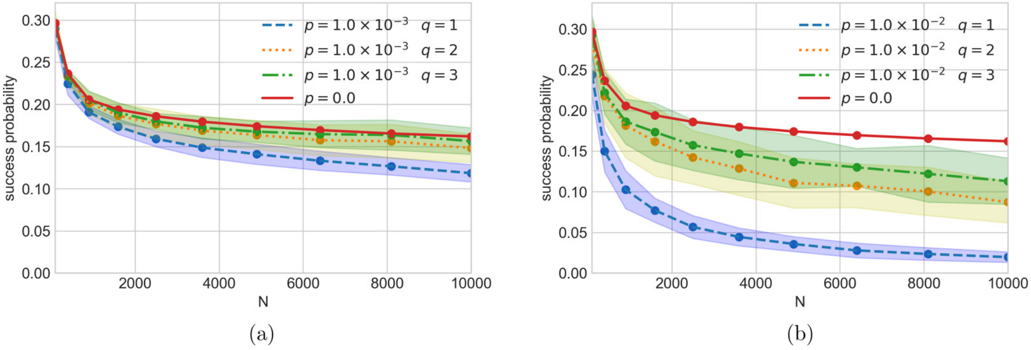


FIG. 12. The average success probability for $n = 10, 20, \ldots, 100$ ($N = n^2$). The dashed blue curve is for $q = 1$, the dotted orange curve is for $q = 2$, the dot-dashed green curve is for $q = 3$, and the solid red line if for $p = 0$. (a) The case for $p = 0.001$. (b) The case for $p = 0.01$.

two polygons are chosen at random. We run the simulations for $p \in \{0.001, 0.01\}$ and $q \in \{1, 2, 3\}$. We obtain the average over 100 runs.

Figure 12 shows the average probability of finding the marked vertex, that is, the success probability, for $n = 10, 20, \ldots, 100$. The case for $p = 0.001$ is depicted in Fig. 12(a), and the case for $p = 0.01$ is shown in Fig. 12(b). The dashed blue curve represents $q = 1$, the dotted orange curve shows $q = 2$, and the dot-dashed green curve shows $q = 3$. The confidence bands for each simulated probability are depicted by the light area around each curve. We can compare them with the solid red curve for $p = 0$. The value of the success probability is closer to the red curve as q increases.

Figure 13 shows the average running time. For $p = 0.001$ [Fig. 13(a)] the curves coincide with the solid red curve for $p = 0$. As p increases, we observe that the bigger q is, the closer we are to the case without decoherence. This can be seen for $p = 0.01$ in Fig. 13(b) and in Fig. 14, where we fix $n = 50$ and consider the range $q = 1, \ldots, 10$. Figures 14(a) and 14(b) depict the average success probability and the average running time, respectively, for $p \in \{0.001, 0.01, 0.1\}$.

V. CONCLUSIONS

We have analyzed decoherence inspired by percolation in the SQW model. We have presented two models of unitary noise by randomly breaking vertices or polygons of the graph. The evolution operators subject to these noises were obtained. The breaking-polygon model is equivalent to breaking some edges of the graph, as it allows us to break edges in a way that the tessellation number of the graph is not increased. On the other hand, breaking an arbitrary edge of the graph may not be that simple. Each polygon is a clique. By removing an edge, the polygon may not be a clique anymore. In order to fulfill the required properties of the SQW model additional tessellations may be needed. This makes the process nontrivial and strictly dependent on the structure of the graph.

Additionally, we have shown the equivalence to the coined quantum walk model when the SQW is obtained from the FCQW model. Breaking vertices can be seen almost as breaking an edge in the coined model. Breaking polygons can be equivalent to modifying the coin or shift operator depending on which tessellation the broken polygon belongs to. Therefore, breaking polygons introduces a different type of decoherence in the coined model.

We have numerically analyzed the effect of breaking vertices and polygons on the SQW on the two-dimensional grid of 4-cliques, which is equivalent to the FCQW on the two-dimensional grid. For both models of decoherence, we have considered a dynamic process in which at each time step a vertex or polygon can be broken according to a constant probability p. The standard deviation of the quantum walk under decoherence was analyzed for different probabilities, and it presents a behavior similar to what happens with the coined quantum walk on the two-dimensional grid with broken links.

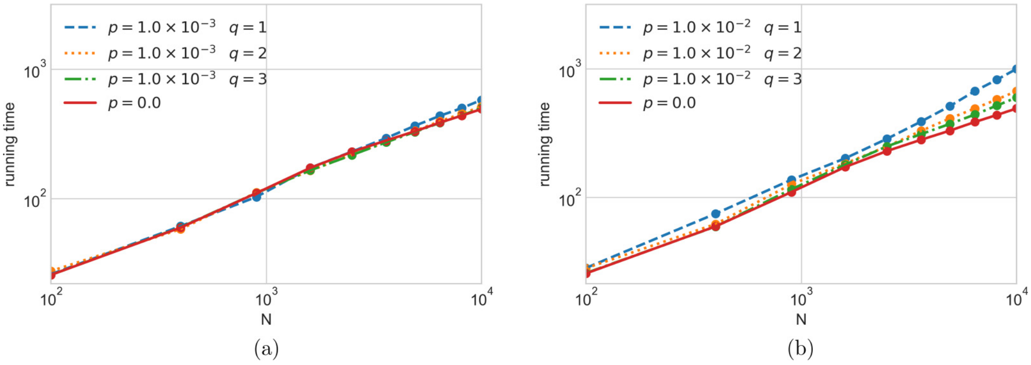
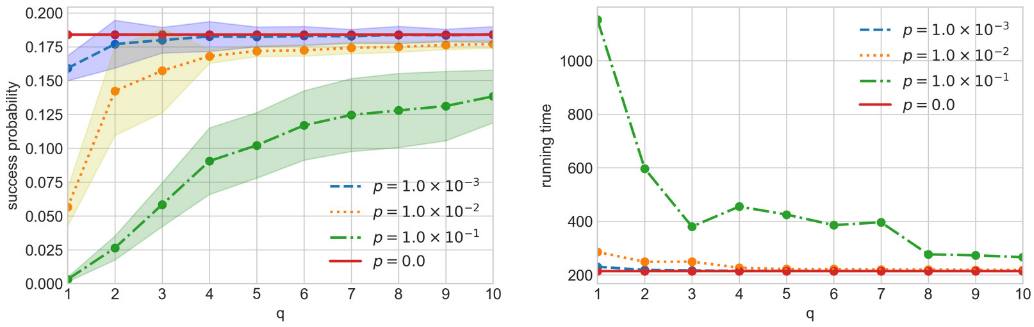


FIG. 13. The average running time for $n = 10, 20, \ldots, 100$ ($N = n^2$). The dashed blue curve is for $q = 1$, the dotted orange curve is for $q = 2$, the dot-dashed green curve is for $q = 3$, and the solid red curve is for $p = 0$. (a) The case for $p = 0.001$. (b) The case for $p = 0.01$.



(a)

(b)

FIG. 14. (a) The average success probability with confidence bands and (b) the average running time for $q = 1, \dots, 10$ and $n = 50$. The solid red curve is for $p = 0$, the dashed blue curve is for $p = 0.001$, the dotted orange curve is for $p = 0.01$, and the dot-dashed green curve is for $p = 0.1$.

The quantum walk loses its quantum behavior as we increase the value of the probability.

The quantum-walk-based search algorithm under the effect of these noises has also been analyzed on the two-dimensional grid of 4-cliques. The plots for the breaking-vertex and breaking-polygon cases are quite similar. The average success probability decreases quite fast as we increase the value of the decoherence probability p. When the probability is quite small, say, $p = 0.001$, the behavior of the decoherent algorithm is still quite close to the algorithm without decoherence. Moreover, we have shown that the algorithm can be more robust against decoherence by expanding the tessellation intersection.

For future research, it would be interesting to analyze decoherence on the SQW on the two-dimensional grid of $4q$-cliques by looking at the probability of finding the walker at single vertices. In this case, we are considering the graph as it is, and it will not be equivalent to the FCQW on the two-dimensional grid. It would also be interesting to investigate other types of decoherence on SQWs.

ACKNOWLEDGMENTS

This work was supported by ERDF Project No. 1.1.1.2/VIAA/1/16/002. F.d.L.M. acknowledges support from CAPES Finance Code 001.

[1] Y. Aharonov, L. Davidovich, and N. Zagury, Quantum random walks, Phys. Rev. A **48**, 1687 (1993).

[2] E. Farhi and S. Gutmann, Quantum computation and decision trees, Phys. Rev. A **58**, 915 (1998).

[3] M. Szegedy, Quantum speed-up of Markov chain based algorithms, in *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science* (IEEE Computer Society, USA, 2004), pp. 32–41.

[4] N. Shenvi, J. Kempe, and K. B. Whaley, A quantum random walk search algorithm, Phys. Rev. A **67**, 052307 (2003).

[5] A. Ambainis, Quantum walk algorithm for element distinctness, in *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science* (IEEE Computer Society, USA, 2004), pp. 22–31.

[6] A. Childs and J. Goldstone, Spatial search by quantum walk, Phys. Rev. A **70**, 022314 (2004).

[7] A. M. Childs, On the relationship between continuous- and discrete-time quantum walk, Commun. Math. Phys. **294**, 581 (2010).

[8] D. W. Berry and A. M. Childs, Black-box Hamiltonian simulation and unitary implementation, Quantum Inf. Comput. **12**, 29 (2012).

[9] R. Portugal, R. A. M. Santos, T. D. Fernandes, and D. N. Gonçalves, The staggered quantum walk model, Quantum Inf. Process. **15**, 85 (2016).

[10] R. Portugal, Establishing the equivalence between Szegedy's and coined quantum walks using the staggered model, Quantum Inf. Process. **15**, 1387 (2016).

[11] G. Coutinho and R. Portugal, Discretization of continuous-time quantum walks via the staggered model with Hamiltonians, Nat. Comput. **18**, 403 (2018).

[12] N. Konno, R. Portugal, I. Sato, and E. Segawa, Partition-based discrete-time quantum walks, Quantum Inf. Process. **17**, 100 (2018).

[13] P. Sadowski, L. Pawela, P. Lewandowska, and R. Kukulski, Quantum walks on hypergraphs, Int. J. Theor. Phys. **58**, 3382 (2019).

[14] R. Portugal and T. D. Fernandes, Quantum search on the two-dimensional lattice using the staggered model with Hamiltonians, Phys. Rev. A **95**, 042341 (2017).

[15] B. Chagas, R. Portugal, S. Boettcher, and E. Segawa, Staggered quantum walk on hexagonal lattices, Phys. Rev. A **98**, 052310 (2018).

[16] R. Portugal, Element distinctness revisited, Quantum Inf. Process. **17**, 163 (2018).

[17] J. Khatibi Moqadam, M. C. de Oliveira, and R. Portugal, Staggered quantum walks with superconducting microwave resonators, Phys. Rev. B **95**, 144506 (2017).

[18] V. Kendon, Decoherence in quantum walks - A review, Math. Struct. Comput. Sci. **17**, 1169 (2007).

[19] A. Romanelli, R. Siri, G. Abal, A. Auyuanet, and R. Donangelo, Decoherence in the quantum walk on the line, Phys. A (Amsterdam, Neth.) **347**, 137 (2005).

[20] A. C. Oliveira, R. Portugal, and R. Donangelo, Decoherence in two-dimensional quantum walks, Phys. Rev. A **74**, 012312 (2006).

[21] B. Kollár, T. Kiss, J. Novotný, and I. Jex, Asymptotic Dynamics of Coined Quantum Walks on Percolation Graphs, Phys. Rev. Lett. **108**, 230505 (2012).

[22] N. B. Lovett, M. Everitt, R. M. Heath, and V. Kendon, The quantum walk search algorithm: Factors affecting efficiency, Math. Struct. Comput. Sci. **29**, 389 (2019).

[23] L. T. Tude and M. C. de Oliveira, Decoherence in the three-state quantum walk, arXiv:2107.09124.

[24] R. A. M. Santos, R. Portugal, and M. D. Fragoso, Decoherence in quantum Markov chains, Quantum Inf. Process. **13**, 559 (2014).

[25] X. P. Xu and F. Liu, Continuous-time quantum walks on Erdös-Rényi networks, Phys. Lett. A **372**, 6727 (2008).

[26] Z. Darázs and T. Kiss, Time evolution of continuous-time quantum walks on dynamical percolation graphs, J. Phys. A **46**, 375305 (2013).

[27] C. Benedetti, M. A. C. Rossi, and M. G. A. Paris, Continuous-time quantum walks on dynamical percolation graphs, Europhys. Lett. **124**, 60001 (2019).

[28] R. Portugal, *Quantum Walks and Search Algorithms* (Springer, New York, 2013).

[29] R. A. M. Santos, The role of tessellation intersection in staggered quantum walks, Nat. Comput. **19**, 843 (2019).

[30] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, Quantum Comput. Inf. **305**, 53 (2002).