Adjacent Vertices Can be Hard to Find by Quantum Walks

N. Nahimovs1*, R. A. M. Santos1**, and K. R. Khadiev2,3***

1Center for Quantum Computer Science, Faculty of Computing, University of Latvia, Riga, LV-1586 Latvia
2OOO Kvantovye Intellektual'nye Tekhnologii, Kazan, 420111 Russia
3Kazan (Volga Region) Federal University, Kazan, 420008 Russia

Received September 11, 2018

Abstract—Quantum walks have been useful for designing quantum algorithms that outperform their classical versions for a variety of search problems. Most of the papers, however, consider a search space containing a single marked element. We show that if the search space contains more than one marked element, their placement may drastically affect the performance of the search. More specifically, we study search by quantum walks on general graphs and show a wide class of configurations of marked vertices, for which search by quantum walk needs $\Omega(N)$ steps, that is, it has no speed-up over the classical exhaustive search. The demonstrated configurations occur for certain placements of two or more adjacent marked vertices. The analysis is done for the two-dimensional grid and hypercube, and then is generalized for any graph. Additionally, we consider an algorithmic application of the found effect. We investigate a problem of detection of a perfect matching in a bipartite graph. We use the found effect as an algorithmic building block and construct quantum algorithm which, for a specific class of graphs, outperforms its classical analogs.

Keywords: quantum algorithms, graph theory, search problem, quantum walks.

DOI: 10.3103/S0278641919010060

1. INTRODUCTION

Quantum computing [1–4] is one of the hot topics in computer science. Researchers have found many problems for which quantum algorithms outperform the best known classical algorithms [5]. One of such problems is search for a vertex in a graph that satisfies some search criteria (usually vertices to be found are referred as marked vertices). One of the most powerful tools for solving this problem is quantum walks.

Quantum walks are quantum counterparts of classical random walks [6]. Similarly to classical random walks, there are two types of quantum walks: discrete-time quantum walks, first introduced by Aharonov et al. [7], and continuous-time quantum walks, introduced by Farhi et al. [8]. For the discrete-time version, the step of the quantum walk is usually given by coin and shift operators, which are applied repeatedly. The coin operator acts on the internal state of the walker and rearranges the amplitudes of going to adjacent vertices. The shift operator moves the walker between the adjacent vertices.

To solve a search problem using quantum walks, we introduce the notion of marked elements (vertices), corresponding to elements of the search space that we want to find. We perform a quantum walk on the search space with one transition rule at the unmarked vertices, and another transition rule at the marked vertices. If this process is set up properly, it leads to a quantum state in which marked vertices have higher probability than the unmarked ones. This method of search using quantum walks was first introduced in [9] and has been used many times since then.

*E-mail: nikolajs.nahimovs@lu.lv.
**E-mail: rsantos@lu.lv.
***E-mail: khadiev@kpfu.ru.
#The article was translated by the authors.
In this paper, we study search by a discrete-time quantum walk on general graphs with multiple marked vertices. We show a wide class of configurations of marked vertices, for which the probability of finding any of the marked vertices does not grow over time. That is, the quantum walk needs $\Omega(N)$ steps and has no speed-up over the classical exhaustive search. We call such configuration as “exceptional-configurations.” They contain two or more marked vertexes.

We start by reviewing a simple example of the two-dimensional grid and show that any pair of adjacent marked vertices forms an exceptional configuration. After that, we extend the proof to general graphs. Namely, we prove that any pair of adjacent marked vertices having the same degree $d$ forms an exceptional configuration. We also prove that the probability of finding a marked vertex in this case is limited by $\Theta(d^2/N)$. Then, we prove that any $k$-clique of marked vertices forms an exceptional configuration. The reason why a configuration of marked vertices is exceptions (i.e. the probability of finding a marked vertex stays close to the initial probability) is because for such configuration the starting state of the algorithm is close to a stationary state. We formulate general conditions for a state to be stationary for a given configuration of marked vertices.

Finally, we consider algorithmic application of the exceptional configurations. We suggest an algorithm for perfect matching detecting in bipartite graphs that can be embedded to two-dimensional grid.

2. TWO-DIMENSIONAL GRID

2.1. Quantum Walk on the Two-Dimensional Grid

Consider a two-dimensional grid of size $\sqrt{N} \times \sqrt{N}$ with periodic (torus-like) boundary conditions. Let $(x, y)$ be nodes of the grid, where $x, y \in \{0, \ldots, \sqrt{N} - 1\}$ The locations of the grid define a set of state vectors, $|x, y\rangle$, which span the Hilbert space, $\mathcal{H}_P$, associated to the position. Additionally, we define a 4-dimensional Hilbert space with the set of states $\{|c\rangle : c \in \{\leftarrow, \rightarrow, \uparrow, \downarrow\}\}$, $\mathcal{H}_C$, associated with the direction.

The evolution of a state of the walk is driven by the unitary operator $U = S \cdot (I_N \otimes C)$, where $S$ is the flip-flop shift operator diffusion transformation

$$C = 2|s_c\rangle\langle s_c| - I_4,$$

where

$$|s_c\rangle = \frac{1}{\sqrt{4}}(|\uparrow\rangle + |\downarrow\rangle + |\leftarrow\rangle + |\rightarrow\rangle).$$

The operator $C$ performs the inversion with respect to the average in dimension 4. Let $|\psi(t)\rangle$ be state of the system after $t$ steps of the algorithm.

The initial state of the algorithm is

$$|\psi(0)\rangle = \frac{1}{\sqrt{N}} \sum_{i,j=0}^{\sqrt{N}-1} |i, j\rangle \otimes |s_c\rangle,$$  \hspace{1cm} (1)

It is the superposition by all vertexes and directions. Note that $|\psi(0)\rangle$ is the only 1-eigenvector of $U$.

The spatial search algorithm uses the unitary operator $U' = S \cdot (I_N \otimes C) \cdot (Q \otimes I_4)$, where $Q$ is the query transformation which flips the sign of marked vertices, that is, $Q|x, y\rangle = -|x, y\rangle$, if $(x, y)$ is marked and $Q|x, y\rangle = |x, y\rangle$, otherwise. Note that $|\psi(0)\rangle$ is a 1-eigenvector of $U'$ but not of $U'$.

In case of one marked vertex, after $O(\sqrt{N \log N})$ steps the inner product $\langle \psi(t)|\psi(0)\rangle$ becomes close to 0. If we measure the state at this moment, we will find the marked vertex with $O(1/\log N)$ probability [10].
Consider a two-dimensional \( \sqrt{N} \times \sqrt{N} \)-grid with two marked vertices \((i, j)\) and \((i + 1, j)\). Let \( \phi^a_{\text{stat}} \) be a state having amplitudes of all basis states equal to \( a \) except for \(|i, j, \rightarrow\rangle\) and \(|i + 1, j, \leftarrow\rangle\), which have amplitudes equal to \(-3a\) (see Fig. 1), that is,

\[
|\phi^a_{\text{stat}}\rangle = \sum_{x,y=0}^{n-1} \sum_c a|x, y, c\rangle - 4a|i, j, \rightarrow\rangle - 4a|i + 1, j, \leftarrow\rangle.
\]

Then, this state is not changed by a step of the algorithm.

**Lemma 1.** Consider a grid of size \( \sqrt{N} \times \sqrt{N} \) with two adjacent marked vertices \((i, j)\) and \((i + 1, j)\). Then the state \( |\phi^a_{\text{stat}}\rangle\), given by Eq. (2), is not changed by the step of the algorithm, that is, \( U|\phi^a_{\text{stat}}\rangle = |\phi^a_{\text{stat}}\rangle\).

**Proof.** Consider the effect of a step of the algorithm on \( |\phi^a_{\text{stat}}\rangle\). The \( Q \) transformation changes the signs of all the amplitudes of the marked vertices. The \( C \) transformation performs an inversion about the average: for unmarked vertices, it does nothing, as all amplitudes are equal to \( a \); for marked vertices, the average is 0, so applying the coin results in sign flip. Thus, \( (I \otimes C)(Q \otimes I) \) does nothing for the amplitudes of the non-marked vertices and twice flips the sign of the amplitudes of the marked vertices. Therefore, we have \( (I \otimes C)(Q \otimes I)|\phi^a_{\text{stat}}\rangle = |\phi^a_{\text{stat}}\rangle\). The \( S \) transformation swaps the amplitudes of nearby vertices. For \( |\phi^a_{\text{stat}}\rangle\), it swaps \( a \) with \( -3a \) and \( -3a \) with \( a \). Thus, we have \( S(I \otimes C)(Q \otimes I)|\phi^a_{\text{stat}}\rangle = |\phi^a_{\text{stat}}\rangle\).

The initial state of the algorithm, given by Eq. (1), can be written as

\[
|\psi(0)\rangle = |\phi^a_{\text{stat}}\rangle + 4a(|i, j, \rightarrow\rangle + |i + 1, j, \leftarrow\rangle),
\]

for \( a = 1/\sqrt{4N} \). Therefore, the only part of the initial state which is changed by the step of the algorithm is \( 4a(|i, j, \rightarrow\rangle + |i + 1, j, \leftarrow\rangle)\). Let us establish an upper bound on the probability of finding a marked vertex.

**Lemma 2.** Consider a grid of size \( \sqrt{N} \times \sqrt{N} \) with two adjacent marked vertices \((i, j)\) and \((i, j + 1)\). Then for any number of steps, the probability of finding a marked vertex \( p_M \) is \( O(\frac{1}{N}) \).

**Proof.** Follows from the proof of Theorem 2 by substituting \( d = 4 \) and \( m = 2N \).

Note that we can construct a stationary state as long as we can tile it by blocks of size \( 1 \times 2 \) and \( 2 \times 1 \). For example, consider \( M = \{(0, 0), (0, 1), (2, 0), (3, 0)\} \) for \( n \geq 3 \). Then the stationary state is given by

\[
|\phi^a_{\text{stat}}\rangle = \sum_{x,y=0}^{n-1} \sum_c a|x, y, c\rangle - 4a|0, 0, \rightarrow\rangle - 4a|0, 1, \leftarrow\rangle - 4a|2, 0, \uparrow\rangle - 4a|3, 0, \downarrow\rangle.
\]

More details on alternative constructions of stationary states for blocks of marked vertices on the two-dimensional grid can be found in [11].
3. GENERAL GRAPHS

3.1. Quantum Walks on General Graphs

Consider a graph $G = (V, E)$ with a set of vertices $V$ and a set of edges $E$. Let $n = |V|$ and $m = |E|$. The discrete-time quantum walk on $G$ has associated Hilbert space $\mathcal{H}^{2m}$ with the set of basis states $\{v, c : v \in V, 0 \leq c < d_v\}$, where $d_v$ is the degree of vertex $v$. Note that the state $|v, c\rangle$ cannot be written as $|v\rangle \otimes |c\rangle$ unless $G$ is regular.

The evolution operator is given by $U = SC$, where $C$ is a coin operator and $S$ is a shift operator. The coin transformation $C$ is the direct sum of coin transformations for individual vertices, i.e. $C = C_1 \oplus \cdots \oplus C_n$, where

$$C_i = 2|s_{d_i}\rangle\langle s_{d_i}| - I_{d_i}, \quad |s_{d_i}\rangle = \frac{1}{\sqrt{d_i}} \sum_{c=0}^{d_i} |c\rangle.$$

$C_i$ is the Grover diffusion transformation of dimension $d_i$. The shift operator $S$ acts in the following way: $S|v, c\rangle = |v', c'\rangle$, where $v$ and $v'$ are adjacent, $c$ and $c'$ represent the directions that points $v$ to $v'$ and $v'$ to $v$, respectively.

The initial state of the algorithm is

$$|\psi(0)\rangle = \frac{1}{\sqrt{2m}} \sum_{v=0}^{n-1} \sum_{c=0}^{d_v-1} |v, c\rangle.$$

It is the superposition by all vertexes and directions. Note that $|\psi(0)\rangle$ is the only 1-eigenvector of $U$.

The spatial search algorithm uses the unitary operator $U' = S \cdot C \cdot Q$, where $Q$ is the query transformation which flips the sign of marked vertices, that is,

$$Q = I - 2 \sum_{w \in M} \sum_{c=0}^{d_w-1} |w, c\rangle\langle w, c|,$$

where $M$ is the set of marked vertexes.

The running time of the algorithm depends on both the structure of the graph as well as the placement of marked vertices.

3.2. Stationary States for General Graphs

Two adjacent marked vertices. Consider a graph $G = (V, E)$ with two adjacent marked vertices $i$ and $j$ with the same degree, that is, $d_i = d_j = d$. Let $|\phi_{\text{stat}}\rangle$ be a state having all amplitudes equal to $a$ except of the amplitude of vertex $i$ pointing to vertex $j$ and amplitude of vertex $j$ pointing to vertex $i$, which are equal to $-(d-1)a$. Figure 2 shows the configuration of the amplitudes in the marked vertices. Then, this state is not changed by a step of the algorithm.

**Theorem 1.** Let $G = (V, E)$ be a graph with two adjacent marked vertices $i$ and $j$ with $d_i = d_j = d$, and let

$$|\phi_{i,j}\rangle = -ad \left(|i, c_{(i,j)}\rangle + |j, c_{(j,i)}\rangle\right),$$

where $c_{(i,j)}$ represents the direction which points vertex $i$ to vertex $j$. Then,

$$|\phi_{\text{stat}}\rangle = a \sum_{v=0}^{n-1} \sum_{c=0}^{d_v-1} |v, c\rangle + |\phi_{i,j}\rangle,$$

is not affected by a step of the algorithm, that is, $U' |\phi_{\text{stat}}\rangle = |\phi_{\text{stat}}\rangle$.

**Proof.** Consider the effect of a step of the algorithm to $|\phi_{\text{stat}}\rangle$. The $Q$ transformation changes the sign of all amplitudes of the marked vertices. The $C$ flip performs an inversion about the average of these amplitudes: for unmarked vertices, it does nothing as all amplitudes are equal to $a$; for marked
vertices, the average is 0, so it results in sign flip. Thus, \( CQ|\phi_{\text{stat}}^a\rangle = |\phi_{\text{stat}}^a\rangle \). The \( S \) transformation swaps amplitudes of adjacent vertices. For \( |\phi_{\text{stat}}^a\rangle \), it swaps \( a \) with \( a \) and \(-(d-1)a \) with \(-(d-1)a \). Thus, we have \( SCQ|\phi_{\text{stat}}^a\rangle = |\phi_{\text{stat}}^a\rangle \).

The initial state of the algorithm \( |\psi(0)\rangle \), given by Eq. (3), can be written as \( |\psi(0)\rangle = |\phi_{\text{stat}}^a\rangle - |\phi_{i,j}^a\rangle \), for \( a = 1/\sqrt{2m} \). Therefore, the only part of the initial state which is changed by a step of the algorithm is \( |\phi_{i,j}^a\rangle \). From this fact, we can establish an upper bound for the probability of finding a marked vertex.

**Theorem 2.** Let \( G = (V, E) \) be a graph with two adjacent marked vertices \( i \) and \( j \) with \( d_i = d_j = d \). Then, the probability of finding of marked vertices is \( p_M = O\left(\frac{a^2}{m}\right) \), where \( m = |E| \).

**Proof.** The only part of the initial state \( |\psi(0)\rangle \) which is changed by the step of the algorithm is \( |\phi_{i,j}^a\rangle = -ad\left(|i, c_{i,j}\rangle + |j, c_{j,i}\rangle\right) \), for \( a = 1/\sqrt{2m} \). Since the evolution is unitary, this part will keep its norm unchanged. In this way, we want to find how big amplitudes can get in order to maximize the value of \( p_M \). This means we want to maximize the function \( 2(d-1)(a+x_1)^2 + 2(-(d-1)a-x_2)^2 \), subject to \( 2(d-1)x_1^2 + 2x_2^2 = |||\phi_{i,j}^a|||^2 = 2a^2d^2 \). Note that \( x_1 \) represents the amplitudes going from the marked vertices to unmarked vertices and \( x_2 \) represents the amplitudes going from one marked vertex to the other. Then, we obtain

\[
p_M \leq 2a^2(2\sqrt{(d-1)d^2} + d(2d-1)) = O\left(\frac{d^2}{m}\right).
\]

One of corollaries of Theorem 2 is that if the degree of the marked vertices is constant or if it does not grow as a function of \( n \), then for large \( n \), the probability of finding a marked vertex will stay close to the initial probability.

**Three adjacent marked vertices.** Now, consider a graph \( G = (V, E) \) with three adjacent marked vertices \( i, j \) and \( k \), that is, a marked triangle. The stationary state for this case will have the amplitudes in the marked vertices as depicted in Fig. 3.

Note that in order to have a stationary state the sum of amplitudes of each marked vertex should be 0, so the action of the coin operator will be a sign flip. By solving the following system of equations:

\[
\begin{align*}
 l_{ij} + l_{ik} &= d_i - 2 \\
 l_{ij} + l_{jk} &= d_j - 2 \\
 l_{ik} + l_{jk} &= d_k - 2,
\end{align*}
\]

we obtain,

\[
l_{ij} = \frac{d_i + d_j - d_k}{2} - 1, \quad l_{ik} = \frac{d_i + d_k - d_j}{2} - 1 \quad \text{and} \quad l_{jk} = \frac{d_j + d_k - d_i}{2} - 1.
\]

**Fig. 2.** Amplitudes of the stationary state for two neighbor vertexes \( i \) and \( j \).

**Fig. 3.** Amplitudes of the stationary state for three neighbor vertexes \( i, j \) and \( k \).
Theorem 3. Let $G = (V, E)$ be a graph with three adjacent marked vertices $i, j, k$; and let
\[
|\phi^a_{i,j,k}⟩ = −a(l_{ij} + 1) (|i, c_{(i,j)}⟩ + |j, c_{(j,i)}⟩) − a(l_{ik} + 1) (|i, c_{(i,k)}⟩ + |k, c_{(k,i)}⟩)
\]
\[-a(l_{jk} + 1) (|i, c_{(j,k)}⟩ + |k, c_{(k,j)}⟩),
\]
where $l_{ij}, l_{ik},$ and $l_{jk}$ are defined in (4). Then,
\[
|\phi^a_{\text{stat}}⟩ = a \sum_{v=0}^{n-1} \sum_{c=0}^{d_v-1} |v, c⟩ + |\phi^a_{i,j,k}⟩,
\]
is not affected by a step of the quantum walk on $G$.

Proof. Similar to Theorem 1.

Note, that in the case of three marked vertices, the configuration is exceptional for any order of vertices.

$k$-clique of marked vertices. Next, we generalize the previous result for any complete subgraph of marked vertices.

Theorem 4. Let $G = (V, E)$ be a graph with $k$ marked vertices $v_1, \ldots, v_k$ forming a $k$-clique. Then it forms an exceptional configuration.

Proof. Let $d_{v_j} = (k - 1) + d'_j$, where $d'_j$ is the number of edges of $v_j$ outside the clique. To construct a stationary state, we need to assign amplitudes to internal edges of the clique, so that the amplitudes in vertex $v_j$ sum up to $d'_j$. Without a loss of generality let $d'_1 < d'_2 < \cdots < d'_k$. We set the amplitude of the edge $(v_1, v_2)$ to $-ad'_1$ and amplitudes of other edges within the clique outgoing from $v_1$ to $0$. By this, we have satisfied the condition for the vertex $v_1$ and reduced the problem from size $k$ to $k - 1$. I.e. now we have a $(k - 1)$-clique with degrees $(d'_2 - d'_1), d'_3, \ldots, d'_k$. Next, we recursively repeat the previous step until we get a 3-clique, which always have an assignment. In this way, we have constructed a stationary state for a $k$-clique of marked vertices.

General conditions. The following theorem gives general conditions under which a state is stationary.

Theorem 5. Let $|ψ⟩$ be a state with the following properties:
1. All amplitudes of the unmarked vertices are equal;
2. The sum of the amplitudes of any marked vertex is $0$;
3. The amplitudes of two adjacent vertices pointing to each other are equal.

Then, $|ψ⟩$ is not changed by a step of the quantum walk, that is, $U|ψ⟩ = |ψ⟩$.

Proof. Item 1 is required in order for the coin transformation to have no effect on the unmarked vertices. Item 2 is necessary so the coin transformation can flip the signs of the amplitudes in the marked vertices. Note that previously the sign of these amplitudes were inverted by the $Q$ transformation, so $C \cdot Q|ψ⟩ = |ψ⟩$. Item 3 is necessary for the shift transformation to have no effect on the state. Therefore, $SCQ|ψ⟩ = |ψ⟩$.

4. THE ALGORITHMIC APPLICATION OF THE EFFECT

It is interesting to find a algorithmic application of the “exceptional configurations” effect. One of the possible applications can be algorithm for perfect matching in bipartite graph detecting. Let us present the problem.

The perfect matching in bipartite graph detecting problem. Let us consider a $G = (V, E)$ graph where $V$ is the set of vertexes and $E$ is the set of edges. Suppose, that we can separate the set of vertices to two parts $V_1$ and $V_2$. The parts are such that $V_1 \subseteq V$, $V_2 = V \setminus V_1$, and there are no edges $e_1, e_2 \in E$ where $e_1 = (v_1, v_2), v_1, v_2 \in V_1$, $e_2 = (v'_1, v'_2), v'_1, v'_2 \in V_2$. The matching is a set $E' \subseteq E$ of edges such that there are no edges from $E'$ with common vertexes. The perfect matching is matching such that $E'$ covers all vertexes. Formally, $V = \{v : v \in \{v_1, v_2\}, e = (v_1, v_2), e \in E'\}$. The task is to detect, does perfect matching exist in the graph $G$ or does not [12].

MOSCOW UNIVERSITY COMPUTATIONAL MATHEMATICS AND CYBERNETICS Vol. 43 No. 1 2019
Let a graph $G$ be called embedded to two dimensional grid $L = (V_L, E_L)$ if we can choose a set of vertexes $V'_L \subseteq V_L$ with the following properties: (i) There is one-to-one mapping for any vertex $v \in V$ to some vertex $V'_L$. (ii) Two vertexes from $V'_L$ are adjacent iff corresponding vertexes in $G$ are adjacent. We call these vertexes marked vertexes.

We present the following algorithm for the problem:

Algorithm.

Step 1. We embed the graph $G$ to $N \times N$-grid, where $N = |V| \log |V|$.

Step 2. We invoke the “Quantum walks” algorithm.

Step 3. If the algorithm found a marked vertex, then there is no perfect matching in the graph. If the algorithm have not found a marked vertex, then there is perfect matching.

Let us explore the time complexity of the algorithm and prove that it works correct.

Theorem 6. The time complexity of Algorithm 1, is $O(N) = O(|V| \log |V|)$. If there is the perfect matching in the graph, then the algorithm found it.

Proof. The time complexity of quantum walks algorithm for such grid is $O(\sqrt{N^2}) = O(N) = O(|V| \log |V|)$.

Assume that there is a perfect matching and it is $E'_L$. Then, by the definition of perfect matching, the matching $E'_L$ separate all vertexes to non-intersected adjacent pairs. Therefore, it is exceptional configuration and the algorithm cannot find marked vertexes; and the algorithm will find non-marked vertex with high probability.

Assume that a perfect matching does not exist in the graph. Additionally, assume that the following statement is true:

$$\sum_{v \in V_{L,1}^{'}} d_v \neq \sum_{v \in V_{L,2}^{'}} d_v,$$

where $v \in V_L'$, $d_v$ is number of adjacent unmurked vertexes of $v$. $V_{L,1}^{'}, V_{L,2}^{'}$ are vertexes from $V_L'$ that belongs to the first and second parts, respectively.

This condition means that there is no exceptional configurations in the graph. Therefore, the algorithm will find a marked vertex and will return right answer.

5. CONCLUSION

In the paper, we explore the behavior of discreet time Quantum walks algorithm for the multi-marked vertexes search problem. We presented a big class of marked vertexes configurations that do not have advantage comparing to the classical brute force algorithm. These configurations are called exceptional configurations. If we mark one of the unmarked vertex, then an exceptional configuration can be converted to regular configuration, and vice verse. Therefore, an additional marked vertex can significantly change the algorithm’s time complexity. There is no similar effect for classical algorithms for the search problem on graphs. In classical case, an additional vertex do not increase the time complexity. We assume that the exceptional configurations effect can be used as subroutine for algorithms. As an example, we suggested the algorithm for perfect matching detecting in bipartite graphs.

REFERENCES

1. M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information (Cambridge Univ. Press, Cambridge, 2010).
2. A. Ambainis, “Understanding quantum algorithms via query complexity,” arXiv:1712.06349 (2017).
3. A. C. C. Say and A. Yakaryilmaz, “Quantum finite automata: a modern introduction,” Lect. Notes Comput. Sci. 8808, 208–222 (2014).
4. R. de Wolf, “Quantum computing and communication complexity,” PhD Thesis (Univ. Amsterdam, Amsterdam, 2001).
5. S. Jordan, Quantum Algorithms Zoo. https://math.nist.gov/quantum/zoo.
6. R. Portugal, Quantum Walks and Search Algorithms (Springer, New York, 2013).
7. Y. Aharonov, L. Davidovich, and N. Zagury, “Quantum random walks,” Phys. Rev. A 48, 1687 (1993).
8. E. Farhi and S. Gutmann, “Quantum computation and decision trees,” Phys. Rev. A 58, 915 (1998).
9. N. Shenvi, J. Kempe, and K. B. Whaley, “A quantum random walk search algorithm,” Phys. Rev. A 67, 052307 (2003).
10. A. Ambainis, J. Kempe, and A. Rivosh, “Coins make quantum walks faster,” in Proceedings of SODA’05 (2005), pp. 1099–1108.
11. N. Nahimovs and A. Rivosh, “Exceptional configurations of quantum walks with Grover’s coin,” in Proceedings of MEMICS 2015 (2015), pp. 79–92.
12. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms-Second Edition (McGraw-Hill, New York, 2001).
13. F. Magniez, M. Santha, and M. Szegedy, “An $O(n^{1.3})$ quantum algorithm for the triangle problem,” in Proceedings of the Sixteenth Annual ACM-SIAM symposium on Discrete algorithms (SODA’05) (ACM–SIAM, New York–Philadelphia, 2005), pp. 413–424.