Simulations of quantum walks on regular graphs

Katalin Friedl

Department of Computer Science and Information Theory
Budapest University of Technology and Economics
Budapest, Hungary
friedl@cs.bme.hu

Viktória Nemkin

Department of Computer Science and Information Theory
Budapest University of Technology and Economics
Budapest, Hungary
nemkin@cs.bme.hu

Abstract: We have developed a simulator program in Python that can execute classical and quantum random walks on regular graphs. The user implements the oracle (a function that returns the adjacency list for a given vertex) and the quantum coin used in the simulation. The software simulates the walk and produces a Latex report file detailing the results. Running several simulations, we compared the behavior of classical and coined quantum walks on some regular graphs and demonstrated the periodicity in a few small special cases. We present such reports with some mathematical explanations.

Keywords: quantum walks, regular graphs, simulation

1 Introduction

Classical random walks are important tools for dealing with large instances of any computational problem that can be formulated as a search. They locally explore the space of possible solutions, iteratively improving on the current candidate via small transformations. It is not guaranteed that the global optimum will be found, however, they can discover good enough approximate solutions and are easy to implement, which can be useful in practical applications. For example, WalkSAT [6] is a popular random walk-based algorithm for testing the satisfiability of CNF formulas. It starts with a random truth assignment, then it repeatedly picks an unsatisfied clause and fixes it by flipping one of its variables until a solution is found or the iteration limit is reached.

Quantum walks [4, 7, 8] are generalized versions of classical random walks on a quantum computer. Since their introduction, the fact that their behaviour is different from their classical counterparts was demonstrated and referenced in many works [4, 5], and they are still being actively researched today. Grover's famous quantum search algorithm [3] can also be viewed as a special case of them[2, 1].

In the literature, there are two types of quantum random walks. The original coined version corresponds to the classical random walk that moves from vertex to vertex on a graph. Since this cannot be used for every graph, there is a more general (and somewhat more complicated) version of quantum walks due to Szegedy [7, 8]. Here we make experiments only with the first, coined version.

2 Quantum walks on regular graphs

In classical random walks on graphs, the only information stored about the system's state is the current position of the walker. This state is updated based on a random choice between the local outgoing edges.

One of the ways classical random walks can be formulated as a quantum algorithm is to implement the random choice as a quantum coin toss. We store the coin's state and update (toss) it using quantum operators, which can result in a superposition of multiple states. The walker moves from a vertex on all of its outgoing edges in superposition (the current coin register determines the amplitudes). In general, the walker is in a superposition of vertices and moves to another superposition.

2.1 Quantum coins

For a d-regular graph, the current state of the coin is represented as a state vector in d-dimensional Hilbert space, each basis state corresponding to an outgoing edge choice. The coin toss is represented as a quantum operator, which is a $(d \times d)$ dimensional unitary matrix, acting on the state vector.

Based on the operator, several types of coins can be defined. The following ones are typically used in quantum walks.

Hadamard coin

The Hadamard coin is the most commonly used quantum coin. It is defined by the Hadamard-matrix, $\mathbf{H}^{\otimes n}$, where $\mathbf{H}$ is

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

An interesting property of this is that starting from one of the basis vector states, measuring after one toss results in a uniform random distribution while measuring after only the second consecutive toss results in the original state with 100% probability since $\mathbf{H}$ is Hermitian.

Grover coin

The Grover coin originates from Grover's search algorithm, where it is applied as the diffusion operator. Let $|D\rangle$ be the uniform state

$$|D\rangle = \mathbf{H}^{\otimes n} |0\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle ,$$

then the Grover coin is the unitary matrix $\mathbf{G} = 2 |D\rangle \langle D| - \mathbf{I}$.

Fourier coin

In contrast to the Hadamard and Grover coins, the Fourier coin can be of any size, not just a power of 2. A Fourier-coin $\mathbf{F}_N$ of size N is defined by the matrix of the Quantum Fourier Transform

$$\mathbf{F}_N = \left[\frac{1}{\sqrt{N}} \omega^{xy} \right]_{x,y},$$

where $\omega = e^{\frac{2\pi i}{N}}$ is an Nth root of unity.

2.2 Quantum walk in 1 dimension

Following [4] the quantum walk on a line is defined using a particle characterised by its position $|x\rangle$ and coin (or spin) state $|s\rangle$. Actually, it is a walk on the circle but when the circle is large enough compared to the number of steps, this can be viewed as a quantum version of the classical random walk on the line.

Coin state In case of the circle, there are two directions to move so the coin state is represented in the two-dimensional space $\mathbb{C}^2$ with basis states $|0\rangle$ and $|1\rangle$. A coin state vector is a unit vector in the form $|s\rangle = s_0 |0\rangle + s_1 |1\rangle$.

Position state At the start of the walk, the particle is at the origin $|0\rangle$. After N steps a classical walker could be in any of the positions between $-N$ and N. For the quantum case, the position state is a unit vector in $\mathbb{C}^{2N+1}$, the basis vectors correspond to the possible positions, denoted by

$$|-N\rangle, |-N+1\rangle, \ldots, |-1\rangle, |0\rangle, |1\rangle, \ldots, |N-1\rangle, |N\rangle.$$

The position state vector is given by

$$|x\rangle = \sum_{i=-N}^{N} x_i |i\rangle.$$

Composite state The composite state of the system with the position and coin state is $|x\rangle \otimes |s\rangle$.

2.3 Evolution

The particle travels on the line based on its current coin state:

- If the current coin state is $|0\rangle$, the particle moves to the left, i.e. from position $|i\rangle$ to position $|i-1\rangle$.
- If the current coin state is $|1\rangle$, the particle moves to the right, i.e. from position $|i\rangle$ to position $|i+1\rangle$.

This step is realised with the unitary matrix $\mathbf{S}$ which operates on the complete state of the system, $|x\rangle \otimes |s\rangle$ and is assembled from a left and a right shift operator acting on $|x\rangle$ and another operator acting on $|s\rangle$ compiled using tensor product.

Left shift operator To move from position $|i\rangle$ to its left to $|i-1\rangle$ the position vector is multiplied with matrix $\mathbf{L}$ that can be expressed in the form

$$\mathbf{L} = |N\rangle\langle -N| + \sum_{i=-(N-1)}^{N} |i-1\rangle\langle i|.$$

It is easy to see that $\mathbf{L}|j\rangle = |j-1\rangle\langle j|j\rangle = |j-1\rangle$, when $-N < j \leq N$, while the first term on the right hand side closes the cycle, achieving $\mathbf{L}|-N\rangle = |N\rangle$, as it was desired.

Right shift operator Similarly, $\mathbf{R} = |-N\rangle\langle N| + \sum_{i=-N}^{N-1} |i+1\rangle\langle i|$ maps $|i\rangle$ to $|i+1\rangle$ and $|N\rangle$ to $|-N\rangle$ performing a right shift.

Shift operator Using matrices $\mathbf{L}$ and $\mathbf{R}$ operating on the position register $|x\rangle$ only, a unitary operator $\mathbf{S}$ can be defined, which operates on the composite state of the system, $|x\rangle \otimes |s\rangle$, executing matrix $\mathbf{L}$ on $|x\rangle$ only when $|s\rangle = |0\rangle$ and matrix $\mathbf{R}$ only when $|s\rangle = |1\rangle$,

$$\mathbf{S} = \mathbf{L} \otimes |0\rangle\langle 0| + \mathbf{R} \otimes |1\rangle\langle 1|.$$

The action of $\mathbf{S}$ on a vector $|x\rangle \otimes |s\rangle$ is $\mathbf{S}(|x\rangle \otimes |s\rangle) = s_0 |x-1, 0\rangle + s_1 |x+1, 1\rangle$.

So when the coin state is $|s\rangle = |0\rangle$ then this $\mathbf{S}$ maps $|x, 0\rangle$ to $|x-1, 0\rangle$ and in the case of $|s\rangle = |1\rangle$ $\mathbf{S}$ maps $|x, 1\rangle$ to $|x+1, 1\rangle$, as intended.

In the quantum setting the coin state can be any mixed state $s_0 |0\rangle + s_1 |1\rangle$ as well. In this case the particle will shift *both* to the left and to the right, at the same time. When measured, the particle can be found in position $|x-1\rangle$ with probability $|s_0|^2$ and in position $|x+1\rangle$ with probability $|s_1|^2$.

Coin operator To replace the classical coin tossing and to inject quantum superposition into the walk, the coin state can be transformed using an arbitrary 2 dimensional unitary matrix between the application of two shift operations. The Hadamard, Grover, and Fourier coins mentioned earlier are commonly used.

For any operator $\mathbf{C}$ on the coin register, the corresponding operator for the composite system that does not modify the position state is $\mathbf{I} \otimes \mathbf{C}$.

Evolution operator Combining the shift operator and the coin operator together, we obtain the following evolution operator, defining one step of the quantum walk on the line. The step consists of flipping the coin once, then applying the shifts

$$\mathbf{U} = \mathbf{S}(\mathbf{I} \otimes \mathbf{C}).$$

3 Quantum walk on regular graphs

Let us have an undirected, connected, regular graph. It will be useful to consider it also as a directed graph having the edges directed in both ways.

In a d-regular graph, the walker must choose from d possible edges to follow at every step. This suggests using a coin with d sides. In the quantum setting the previous 2-dimensional coin state is replaced by a d dimensional state vector, the basis vectors are $|0\rangle, |1\rangle, \ldots, |d-1\rangle$ corresponding to the different choices.

The coin operator is a unitary matrix $\mathbf{C} \in \mathbb{C}^{d \times d}$. The evolution operator formally looks the same as for the line, $\mathbf{U} = \mathbf{S}(\mathbf{I} \otimes \mathbf{C})$

To generalize the shift operator, the previous left and right shifts are replaced by d transition matrices with nonnegative elements,

$$\mathbf{S} = \mathbf{S}_0 \otimes |0\rangle \langle 0| + \mathbf{S}_1 \otimes |1\rangle \langle 1| + \cdots + \mathbf{S}_{d-1} \otimes |d-1\rangle \langle d-1|,$$

where $\mathbf{S}_0 + \mathbf{S}_1 + \cdots + \mathbf{S}_{d-1}$ is the adjacency matrix of the graph (as it was in the previous case, where $\mathbf{L} + \mathbf{R}$ was the adjacency matrix of the circle).

Since in a quantum walk the operators have to be unitary, $\mathbf{S}$ has to be unitary. This gives the following condition for the good decompositions of the adjacency matrix:

Theorem 1 *Let $\mathbf{S}_0, \ldots, \mathbf{S}_{d-1}$ be matrices with nonnegative elements and assume that $\sum_{i=0}^{d-1} \mathbf{S}_i$ is the adjacency matrix of a d-regular graph. The operator $\mathbf{S} = \sum_{i=0}^{d-1} \mathbf{S}_i \otimes |i\rangle \langle i|$ can be a shift operator of a quantum walk on the graph if and only if the $\mathbf{S}_i$ are permutation matrices.*

□

Corollary 2 *In the previous Theorem all the $\mathbf{S_i}$ are symmetric if and only if they correspond to a d coloring of the edges of the undirected graph, $\mathbf{S_i}$ is the adjacency matrix of the ith color class.*

□

Although usually this decomposition, based on edge coloring is mentioned, there are (nonsymmetric) possibilities, even when coloring the edges needs more than d (namely $d+1$) colors, since

Fact 3 *The adjacency matrix of any d-regular graph can be obtained as a sum of permutation matrices.*

4 Properties of quantum walks

Similarly to classical random walks, the effect of several steps in a quantum walk can be described by a power of the matrix representing one step. However, quantum walks behave differently than classical walks, since the matrix $\mathbf{U} = \mathbf{S}(I \otimes \mathbf{C})$ is unitary, therefore all of its eigenvalues have unit length, they cannot diminish. Furthermore, when all the eigenvalues are Mth roots of unity, then the walk is periodic by M.

The eigenvalues in some special cases can be computed from smaller matrices. Let $\lambda(\mathbf{A})$ denote the spectra of the operator $\mathbf{A}$. Then it is easy to see the following

Theorem 4 *Let* $\mathbf{U} = \mathbf{S}(I \otimes \mathbf{C})$ *where* $\mathbf{S} = \sum_{j=0}^{d-1} \mathbf{S}_j \otimes |j\rangle \langle j|$. *Assume that the* $\mathbf{S}_j$ *have a common eigenvector basis, i.e.* $\mathbf{S}_j v_k = \lambda_{j,k} v_k$, *where* $0 \leq j < d$ *and* $0 \leq k < n$. *Then* $\lambda(\mathbf{U}) = \bigcup_{k=0}^{n-1} \lambda(\mathbf{\Lambda}_k \mathbf{C})$, *where the* $\mathbf{\Lambda}_k$ *are* $d \times d$ *diagonal matrices formed from the* $\lambda_{j,k}$.

□

This shows that although the matrix of the walk has size $nd \times nd$ its eigenvalues can be computed from n matrices of sizes $d \times d$.

The condition of the Theorem is fulfilled, for example, when the $\mathbf{S}_j$ commute.

An interesting special case is when the graph is a Cayley graph of an Abelian group. Let Γ be an Abelian group, and $B \subseteq \Gamma$ be a symmetric generating system (i.e. $g \in B$ implies $g^{-1} \in B$). The vertices of the Cayley graph are the elements of Γ and there is an edge from a to b if $b = ag$ for some $g \in B$. This is a regular graph, the adjacency matrix of edges belonging to a fixed $g \in B$ form a permutation matrix, they provide a good decomposition for the adjacency matrix of the Cayley graph. It is known that they have common eigenvectors (formed from the characters of the group), the eigenvalues are values of characters.

Based on this, in some special cases, the eigenvalues of the quantum walk are not too difficult to compute.

For example, in the case of the circle when $n = 4$ then the eigenvalues are 8th roots of unity, when $n = 8$, then 24th roots of unity. This means that the walk is periodic in these cases. For a general n the eigenvalues $e^{i\varphi}$ are such that $\sqrt{2} \sin \varphi = \sin \frac{2\pi a}{n}$ holds ($a = 0, 1, \ldots, n-1$).

The rest of the section shows some results of our simulations.

4.1 Walks on a circle

The first walk to be reviewed is the 1-dimensional walk, it is a walk on a circle (with 128 vertices, numbered along the perimeter). Since this is a 2-regular graph, a 2-dimensional coin is used.

The 2-dimensional Hadamard and Fourier coins are identical, while the 2-dimensional Grover coin is just an X gate, which means the walker stays around to the starting position at all times.

In the following figures, we can see the changes in the probability distribution during the walk. The x axis contains the vertices, and the y axis contains the steps. The walker starts from the center (vertex 64), and in the classical case, multiple runs are done to arrive at a probability distribution, while in the quantum case, a single walker is enough, as it spreads in superposition over the graph.

The ballistic nature of the walk can be seen from steps 0 to around 100, where the bright red diagonals represent a strong probability concentration moving away from the origin. When the probability bumps reach the sides, they cross over and travel towards the center at the opposite side. We can also see secondary, tertiary, and further red lines traveling alongside the main ones. These reach cross over later, which results in a weaved pattern.

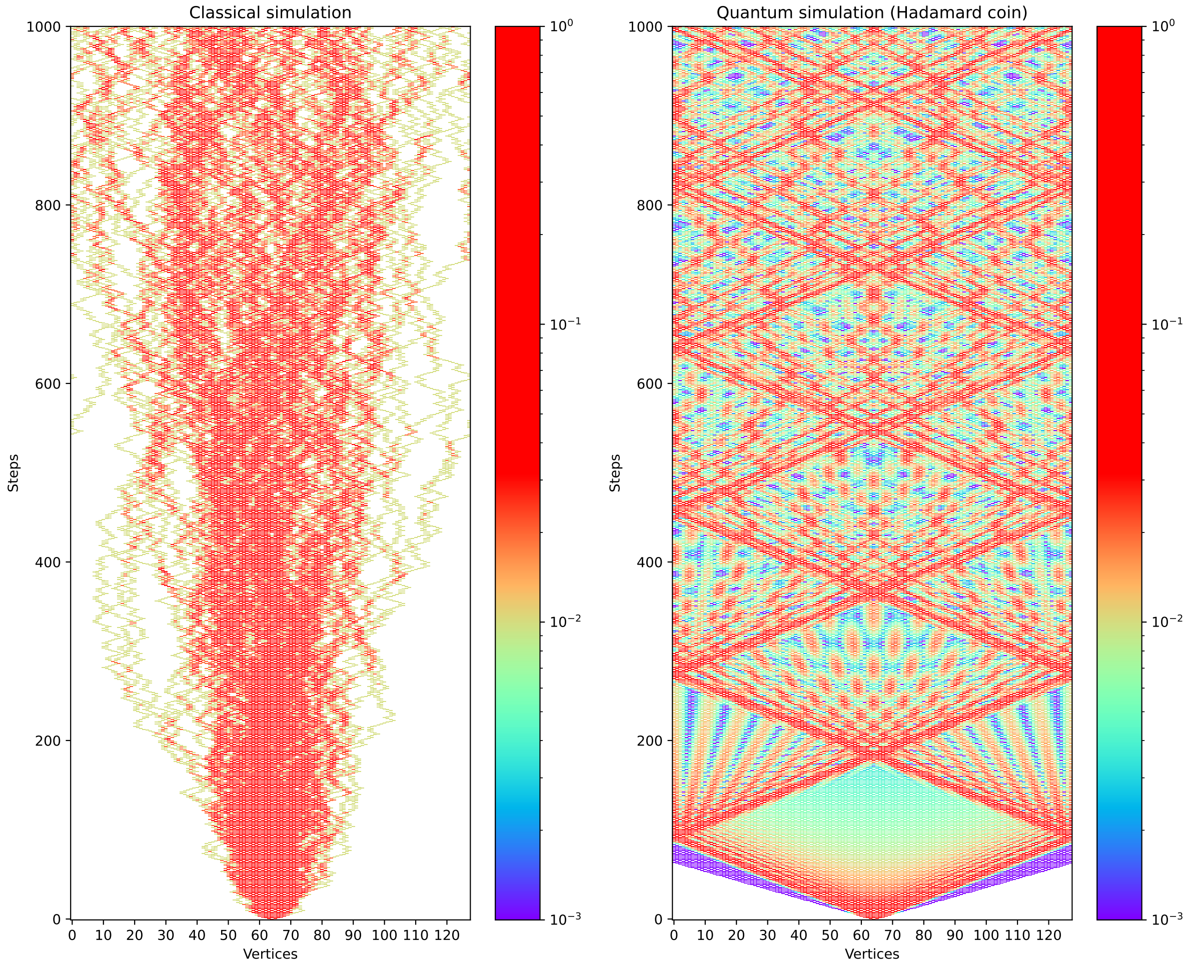

Figure 1: Probability distribution of walks on a larger circle of 128 vertices

4.2 Walks on the grid

The second graph reviewed is the 2-dimensional grid (torus), with $4 \times 4 = 16$ vertices, indexed in row-major order. The vertices are horizontally and vertically connected, the outer vertices connecting to the vertex at the opposite end of the same row or column.

The following 4 images contain the classical, the quantum Hadamard, the quantum Grover and the quantum Fourier walks on the grid. The walker starts from the center (vertex 8). The classical walk quickly spreads over the graph since all vertices are close to each other (as opposed to the line, where the maximum distance is large).

In the quantum case, using the Hadamard and Grover coins, an interesting quality can be distinctly observed: these quantum walks are periodic. On the 4×4 grid using the Hadamard coin, the periodicity is 40 steps, while using the Grover coin, it is 12 steps.

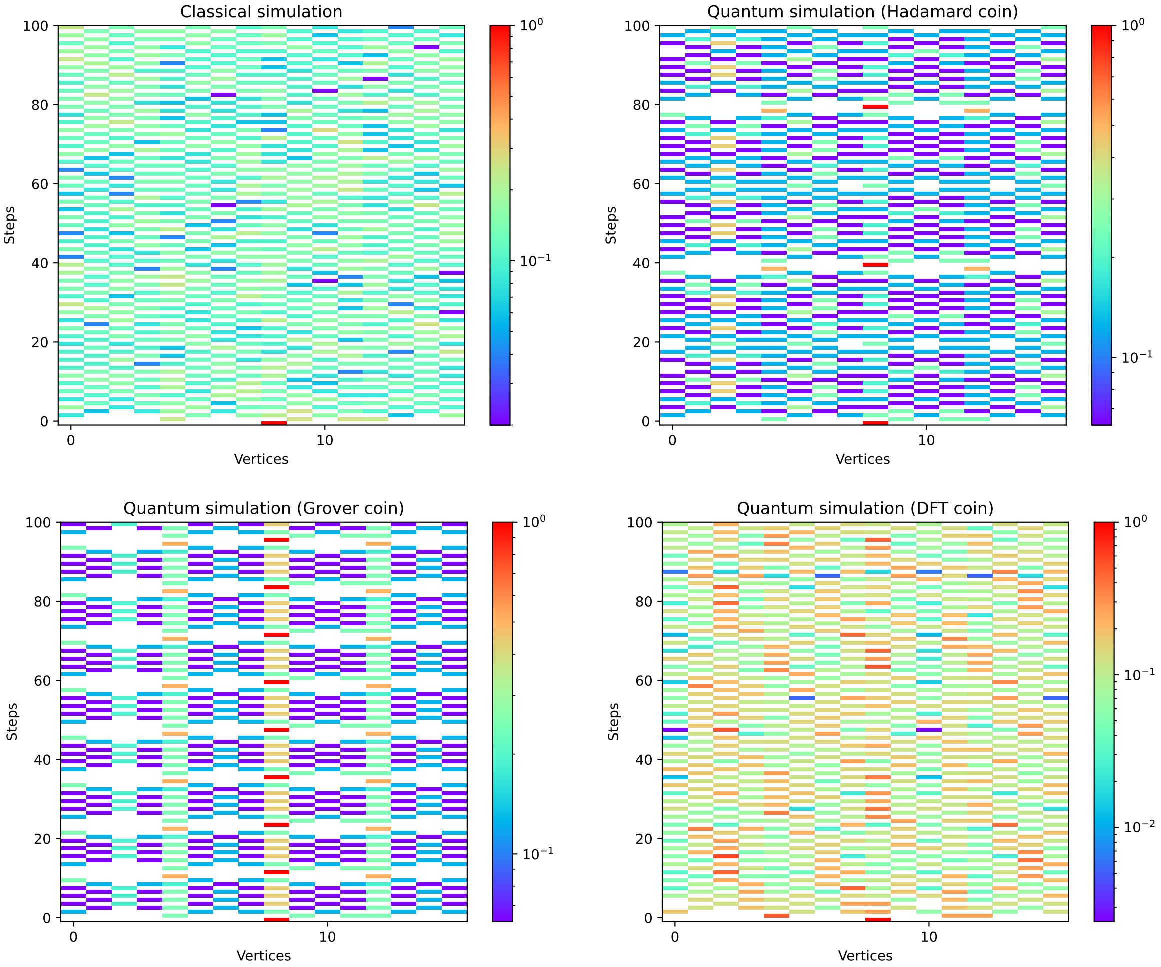

Figure 2: Probability distribution of walks on the grid (with horizontal / vertical steps)

4.3 Walks on hypercube

The third graph reviewed is the 4-dimensional boolean hypercube (with $2^4 = 16$ vertices).

On the 4 dimensional hypercube using the Hadamard coin, the periodicity is 24 steps, using the Grover coin, it is 12 steps while using the Fourier coin it turns out to be 48 steps.

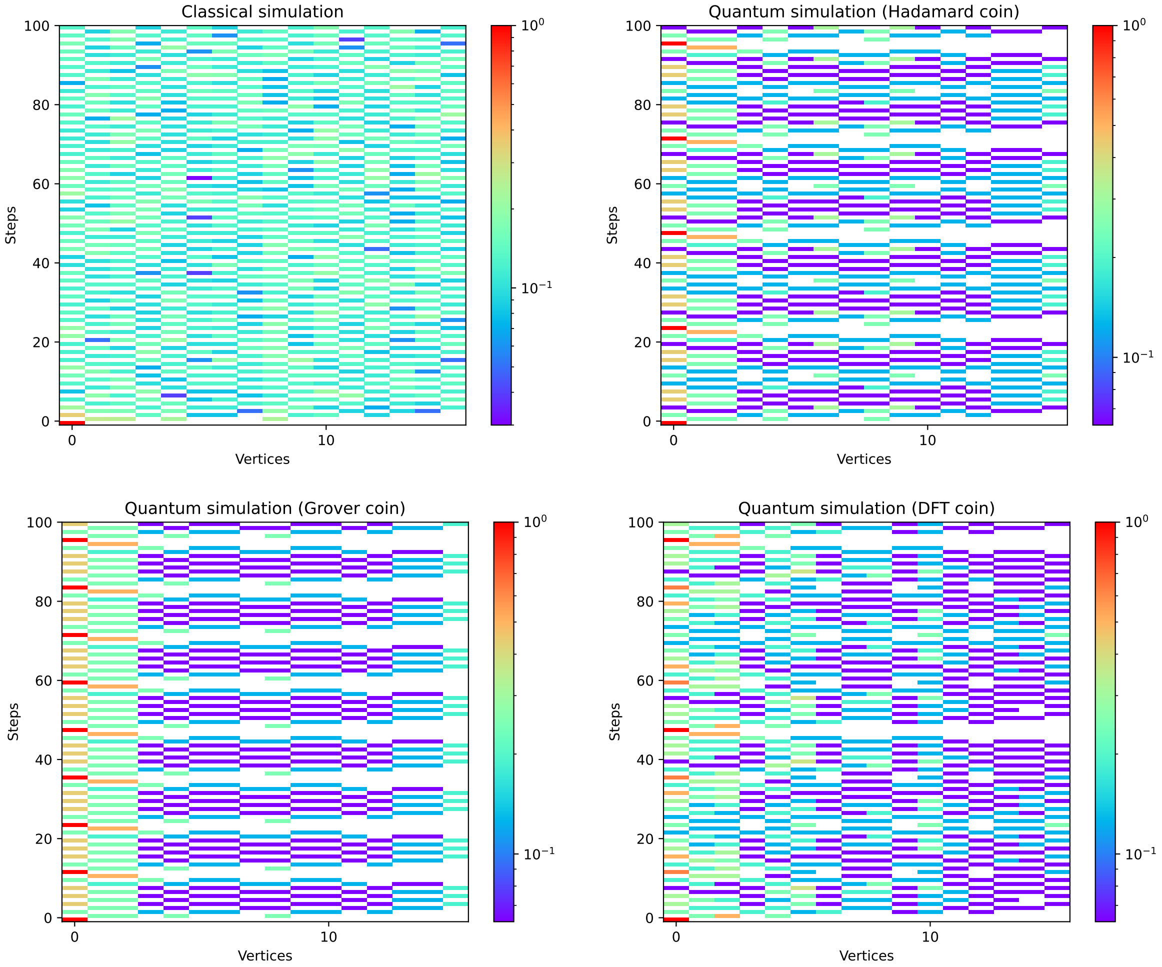

Figure 3: Probability distribution of walks on the hypercube

References

[1] A. Ambainis, Quantum walks and their algorithmic applications, *International Journal of Quantum Information* **1(4):507–518** (2003)

[2] F. Magniez, A. Nayak, J. Roland, and M. Santha, Search via Quantum Walk, *SIAM Journal on Computing* **40(1):142–164** (2011)

[3] L. K. Grover, A Fast Quantum Mechanical Algorithm for Database Search, *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* **pp. 212–219** (1996)

[4] J. Kempe, Quantum random walks: An introductory overview, *Contemporary Physics* **44(4):307–327** (2003)

[5] R. Portugal, Quantum Walks and Search Algorithms, *Springer International Publishing* **Quantum Science and Technology** (2018)

[6] B. Selman, H. Kautz, and B. Cohen, Local Search Strategies for Satisfiability Testing, *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge* **26** (1999)

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

[8] R. de Wolf, Quantum Computing: Lecture Notes, *arXiv:1907.09415* **v4** (2022)