



UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE E TECNOLOGIE

DIPARTIMENTO DI FISICA

Corso di Laurea triennale in Fisica (L-30)

**Quantum walks with time-dependent Hamiltonians
and their application to the search problem on graphs**

Relatore : **Prof. Matteo G.A. Paris**
Correlatore: **Prof. Stefano Olivares**
Correlatrice: **Dott.sa Claudia Benedetti**

Tesi di Laurea di:
Matteo Garbellini
Matricola 885615
Anno Accademico 2019/2020

Contents

Introduction	4
1 Preliminaries	6
1.1. Introduction to graph theory	6
1.1.1 Cycle graph	8
1.1.2 Complete graph	9
1.2. Quantum walks	9
1.3. Grover's quantum search	11
1.4. Search by quantum walks	13
1.4.1 Search on the complete graph	15
1.5. Search by adiabatic evolution	16
1.5.1 Adiabatic theorem	16
1.5.2 Global adiabatic evolution	18
1.5.3 Local adiabatic evolution	19
1.6. Adiabatic-quantum walks search	22
2 Quantum walks with time-dependent Hamiltonians	25
2.1. Search with time-dependent Hamiltonian	25
2.1.1 Time-dependent quantum walks	26
2.1.2 Interpolating schedule $s(t)$	27
2.1.3 Multiple runs for one search	28
2.2. Selected topologies: cycle and complete graph	29

2.3. Search, localization and robustness	29
2.3.1 Search vs Localization	29
2.3.2 Robustness	30
2.4. Results for the cycle graph	32
2.4.1 Time-independent benchmarks	32
2.4.2 Time-dependent results	34
2.4.3 Comparison: localization	36
2.4.4 Comparison: search	38
2.4.5 Comparison: robustness	46
2.5. Results for the complete graph	48
2.5.1 Comments on the placement of γ	48
2.5.2 Probability distribution and qualitative robustness	49
Conclusions	52
Bibliography	56

Introduction

Grover's quantum search algorithm has become one of the most celebrated algorithms in quantum computing and quantum information. It enables to outperform classical algorithms by achieving a quadratic speed up in the unstructured database search. Moreover, the search algorithm is *general* in the sense that it can be applied to a broad range of scenarios beyond the database search [9], from the *collision problem* [1] to solving NP-complete problems [6].

The original Grover's algorithm was formulated in a time when the quantum circuit model was the mainstream tool in quantum computation, consisting in a series of quantum gates applied in discrete time [11]. In 1998 Farhi and Gutmann introduced the idea of analog quantum computing, where the system evolves in continuous-time following the Schrodinger equation [5]. Later on, when quantum walks were popularized as algorithmic tools and the adiabatic evolution with time-dependent Hamiltonians emerged, the search problem was soon investigated [3, 4]. Therefore, when new models for quantum computing are introduced, new formulations of Grover's algorithm soon follow. It came as a natural consequence the idea of comparing its different formulations, highlighting their differences and similarities [11].

In this thesis we study the application of quantum walks with time-dependent Hamiltonian to the search problem on graph.

We compare the standard time-independent quantum walks search with the time-dependent one, with the goal of understanding if this implementation can lead to an improvement in performance for selected graphs. In particular we study its application to the cycle graph, where the search problem is not solved with the standard quantum walk of Farhi and Gutmann, and for completeness we give some results for the complete graph, for which the quantum walk search is a perfect search. In order to do so we compare the two approaches in terms of search, localization and a measure of robustness.

Our work can be summarized as follows:

- In Chapter 1 we recall the minimum theoretical knowledge necessary to follow the work done in this thesis. We review some basic notion of graph theory, quantum walks and the main characteristics of the graph considered. We then introduce the quantum search problem as firstly posed by Grover, followed by its quantum walks implementation. Then we discuss the adiabatic theorem and its application to the search problem, focusing in particular on the difference between global and local adiabatic evolution. Lastly we show that an adiabatic-quantum walks search algorithm is not possible.
- In Chapter 2 we study quantum walks with time-dependent Hamiltonians, focusing on the application to the search problem on graph. In particular we focus on the cycle graph and the complete graph. The goal of this section is to determine if a time-dependent Hamiltonian, inspired by the adiabatic implementation, can bring any advantages to the search problem on selected graphs.

Finally, conclusions and future perspective complete this thesis.

Preliminaries

In this chapter we present the basic theoretical knowledge necessary to understand this thesis. We begin by introducing graph theory, random walks and quantum walks. We then discuss the quantum search problem firstly introduced by Grover and the quantum walks implementation for the unstructured search by Childs and Goldstone. Then we present the adiabatic theorem and its application to the adiabatic quantum search, studying the global and local adiabatic evolution. Lastly we look at the differences between the quantum walks approach and the local adiabatic evolution search by Roland and Cerf, which sets the basis for our work.

1.1. Introduction to graph theory

A graph G is defined as a ordered pair (V, E) , where V is a set of vertices and E is a set of edges, that represents the connection between any two pair of vertices. A vertex is usually indicated by the cursive letter j , and the corresponding edge connecting j to i is given by (j, i) . Given a set V of dimension N , we refer to a particular vertex with its index, namely $j = 1, 2, \dots, N$.

A graph can be characterized by many properties. Throughout our work we will only consider *simple graphs* characterized by being *undirected*, namely the edges E are symmetric, without self loops such that $(j, j) \notin G$ and having no multiple equivalent edges. Additionally we require the graph to be *connected*, i.e. each vertex can be reached by any other following a path through the available edges. We then define the *vertex degree* d_j , that represents, given a vertex j , the number

of edges that are incident to the vertex j ¹.

If indeed any two vertices i and j are connected by an edge (i, j) we define them as *adjacent*, and from this we can construct an analytical representation of a graph, given by an N -dimensional square matrix called *adjacency matrix*, usually referred as A . The adjacency matrix is defined as following ²:

$$A_{ij} = \begin{cases} 1 & \text{if } (i, j) \in G \\ 0 & \text{otherwise} \end{cases} \quad (1.1)$$

which represents the connectivity of the graph.

Furthermore, we can introduce a diagonal matrix D that encodes the informations of the vertex degrees for a particular graph G . Calling the N vertices of the graph as $j = 1, 2, \dots, N$ the matrix D is defined as:

$$D = \text{diag}(d_1, \dots, d_N) \quad (1.2)$$

where d_j is the degree of the vertex j . In this particular context a natural operative basis arises, in which one can associate to each ordered vertex of the graph a vector of the standard basis of the N -dimensional vector space.

In order to study the dynamics of the system we introduce the *Laplacian matrix* L , also known as the *discrete Laplacian operator*. It is defined as

$$L = D - A \quad (1.3)$$

where D is the diagonal degrees matrix and A is the adjacency matrix. The discrete Laplacian operator for the finite, undirected, simple and connected graphs can be characterized by the following properties:

- L is symmetric given that both D and A are symmetric
- the sum of all elements over a row/column equals to zero

¹For an undirected graph the degree does not depend on the edges incident from or to the selected vertex, while generally this is not true

²If we had to define the adjacency matrix for an unspecified graph the value of A_{ij} is not necessarily equal to one, but a general a_{ij} since it takes into account the possibilities of self loops and multiple equivalent edges. For the simple graphs considered in this thesis the given definition suffices.

- it has a null eigenvalue which corresponds to the eigenvector $|\Psi_0\rangle = \frac{1}{\sqrt{N}}(1, 1, \dots, 1)$

As previously mentioned we can associate every vertex to a basis vector of the N -dimensional vector space. Similarly we can expand this notation to an N -dimensional Hilbert space, where we can use the following Dirac notation to describe the vertex. This formalism allows us to write the Laplacian matrix in convenient quantum mechanical form using the projectors $|j\rangle\langle i|$

$$|1\rangle = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad |2\rangle = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \quad |N\rangle = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ N \end{bmatrix} \quad (1.4)$$

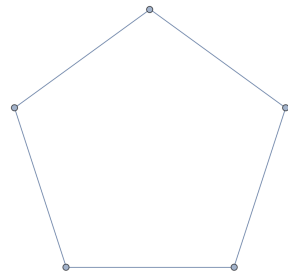
In the following paragraphs we give a brief description of the graphs that will be consider throughout the work.

1.1.1 Cycle graph

A N -dimensional cycle graph $Cy(N)$ is a monodimensional structure with periodic boundary conditions $|N+1\rangle = |N\rangle$. The Laplacian is given by

$$L = 2 \sum_{k=1}^N |k\rangle\langle k| - \sum_{k=1}^{N-1} |k\rangle\langle k+1| - \sum_{k=2}^N |k\rangle\langle k-1| - |N\rangle\langle 1| - |1\rangle\langle N| \quad (1.5)$$

A pictorial representation of a cycle graph is given in Figure 1.1 with the corresponding Laplacian matrix.



(a)

$$\begin{pmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{pmatrix}$$

(b)

Figure 1.1: Pictorial representation of a cycle graph with 5 nodes (a), and the matricial representation for the Laplacian of $Cy(5)$

1.1.2 Complete graph

A graph with N vertices is said to be complete if every node is adjacent to all the other $N - 1$ nodes, thus representing a finite bidimensional structure. Its Laplacian matrix is given by:

$$L = (N - 1) \sum_{j=1}^N |j\rangle\langle j| - \sum_{k \neq j} |j\rangle\langle k| \quad (1.6)$$

A pictorial representation of a complete graph is given in Figure 1.2. We will refer to this type of graph with $C(N)$.

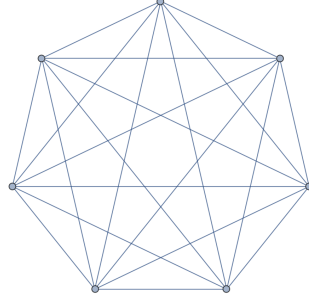


Figure 1.2: Pictorial representation of a complete graph with $N=7$

1.2. Quantum walks

The continuous-time quantum walk (CTQW) is the direct analogue of the classical continuous-time random walk (CTRW). We begin by considering the classical one, expanding later on the quantum mechanical counterpart.

A continuous-time random walk is a stochastic process that describes the random motion of a particle on a mathematical structure, which in our scenario is a graph G . Let j be a node of a graph G , such that the initial state of the system is $|j\rangle$ ³. Then, we denote the transition probability of the walker to go from node j to a node k in a time t with $p_{k,j}(t)$. The state after time t is given by $|j;t\rangle$, such that the overlap with node k is $\langle k|j;t\rangle = p_{j,k}(t)$.

The dynamics that results in the state $|j;t\rangle$ follows from probability of transitioning from a node to another, also called *transition rates*. In particular these

³With abuse of notation we represent both the classical and the quantum mechanical vertex with the Dirac notation $|j\rangle$

transition rates are the components of the so-called *transfer matrix* T , namely $T_{ij} = \langle k|T|j \rangle$. If we assume a Markovian process, the following master equation defines the CTRW evolution:

$$\frac{d}{dt}p_{k,j}(t) = \sum_l T_{kl}p_{l,j}(t). \quad (1.7)$$

In the simplest case, where the transitions rates for all edges are equal, the transfer matrix is closely related to the Laplacian matrix through :

$$T = -\gamma L \quad (1.8)$$

where γ is the transition rate. The solution of eq. (1.7), along with the normalization constrains $\sum_{k=1}^N p_{k,j}(t) = 1 \forall t$, is given by

$$p_{ij}(t) = \langle k|e^{Tt}|j \rangle = \langle k|e^{-\gamma L t}|j \rangle. \quad (1.9)$$

Turning to quantum mechanics the evolution of any physical system obeys the Schroedinger equation, and QWs represent no exception. The dynamics of the CTQW is governed by a specific Hamiltonian H , such that the Schroedinger equation for the transition *amplitudes* $\alpha_{i,j}(t)$ is given by

$$\frac{d}{dt}\alpha_{i,j}(t) = -i \sum_l H_{k,l} \alpha_{l,j}(t) \quad (1.10)$$

where H is the Hamiltonian of the system, and for semplicity we assume $\hbar = 1$. The formal solution of such differential equation is given by

$$\alpha_{l,j}(t) = \langle k|e^{-iHt}|j \rangle \quad (1.11)$$

where e^{-iHt} is the quantum mechanical time-evolution operator. We immediately notice the similar structure of equations (1.7) and (1.11), with the only difference - apart from the imaginary unit - that the first is a differential equation for transition probabilities while the latter allows to compute transition amplitudes. The similarity is further pushed by Farhi and Gutmann, proposing to identify the Hamiltonian H of the system with the negative of the classical transfer matrix T

[2], which as we have seen previously is the Laplacian of the graph ⁴:

$$H = -T = L \quad (1.12)$$

Therefore the Laplacian matrix completely determines the evolution of the quantum system as well as the classical scenario.

1.3. Grover's quantum search

In 1997 Lov K. Grover addressed the search problem through a quantum mechanical algorithm [7]. The search problem itself can be formulated in the following way: given an unsorted database containing N items, find one particular item that satisfied a certain condition. Once an item is examined, it is possible to determine whether it represents the solution or not in just one step. Classically, the most efficient algorithm has to check each and every item in the database individually. If the item checked satisfies the condition the process stops, otherwise it will continue to examine the remaining items until the solution is found. On average, the algorithm will have to check $N/2$ items before finding the desired one.

On the other hand, Grover's quantum mechanical approach takes advantage of the *superposition* of states in a quantum system, and by having input and output in such superposition it can find the desired item in $O(\sqrt{N})$ *quantum mechanical steps*, instead of $O(N)$ classical steps. As we shall see later, these quantum mechanical steps consist of rotations due to a unitary operator called *Grover operator*.

Let us look at the quantum search algorithm more in detail, in particular setting the stage for the search algorithm in terms of an *oracle*, which plays a central role in Grover's search and throughout our work. Additionally, this allows us to present a very general description of the search procedure, and a geometric way to visualize its action [9].

Suppose we want to search through a search space of N elements, and let us consider the indices of such elements - instead of the actual elements - that are numbers from 0 to $N - 1$. In order to easily store the index in n bits we consider $N = 2^n$, and assume that the particular search problem has only one solution.

⁴Please note that in this particular scenario the transition amplitude γ is set to one, and assumed to be equal for all vertex transition.

We can now easily represent one instance of the search, i.e. checking if the item satisfies a particular condition, with a function $f(x)$, where x is the index integer ranging from 0 to $N - 1$. The function is such that if x represents the solution to the search problem $f(x) = 1$, and $f(x) = 0$ if x is not a solution. In particular we suppose that we are supplied by a quantum *oracle* U_w with the ability to *recognize* the solution of the search problem. The action of the oracle U_w is to *mark* the solution by applying a phase, namely $|x\rangle \xrightarrow{U_w} (-1)^{f(x)}|x\rangle$.

In a simplified manner, the quantum search algorithm proposed by Grover consists of repeated applications of a quantum subroutine known as the *Grover iteration* or *Grover operator* U_G defined as:

$$\begin{aligned} U_G &= U_s U_w \\ &= (2|s\rangle\langle s| - I)U_w \end{aligned} \quad (1.13)$$

where I the identity matrix and $|s\rangle$ is the initial state given by the superposition of all the items states:

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_x^N |x\rangle \quad (1.14)$$

The question that arises is what does this Grover operator do. We can show that it can be regarded as a *rotation* in the 2-dimensional space spanned by the beginning vector $|s\rangle$ and the vector $|w\rangle$ representing the solution to the search problem. We can then define the normalized state $|\alpha\rangle$ and $|\beta\rangle$:

$$|\alpha\rangle = \frac{1}{\sqrt{N-1}} \sum_{x \neq w} |x\rangle \quad (1.15)$$

$$|\beta\rangle = |w\rangle \quad (1.16)$$

representing respectively, the superposition of all states that are not a solution and the target state. Therefore the initial state $|s\rangle$ can be re-expressed as:

$$|s\rangle = \sqrt{\frac{N-1}{N}} |\alpha\rangle + \sqrt{\frac{1}{N}} |\beta\rangle \quad (1.17)$$

The effect of U_G can be understood by realizing that the oracle U_w performs a *reflection* of $|s\rangle$ about the vector $|\alpha\rangle$ in the plane defined by $|\alpha\rangle$ and $|\beta\rangle$. Then, U_s performs another reflection about the vector $|s\rangle$. As can be seen in Figure 1.3, the product of these two reflections is a rotation. By applying the Grover operator k times we can rotate the initial state vector $|s\rangle$ close to $|\beta\rangle$, that is, the solution to

the search problem. The number of necessary iterations - for which the derivation is beyond the scope of this thesis - is given by $\frac{\pi}{4}\sqrt{N}$. Therefore we can identify a *quantum mechanical step* with the action of the Grover operator U_G .

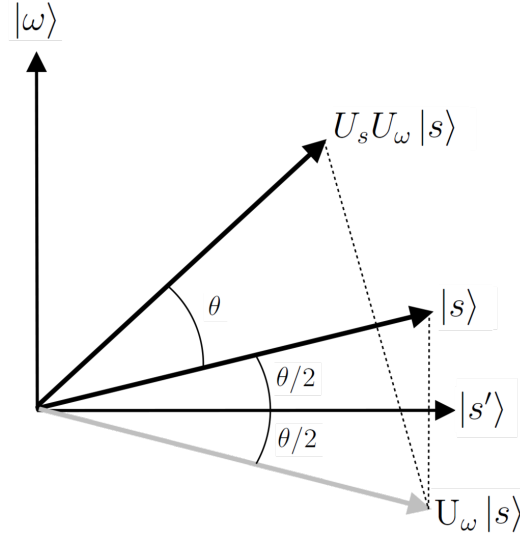


Figure 1.3: **Action of the Grover operator U_G :** The figure shows the action of the Grover operator $U_G = U_s U_w$. Starting from the initial state $|s\rangle$ the action of the Oracle U_w reflects it about the vector $|\alpha\rangle$. Then, the action of U_s is an additional reflection about $|s\rangle$. The application of the Grover operator for $\pi/4\sqrt{N}$ times rotates the initial state bringing it close to $|\beta\rangle$, that is, the solution of the search problem.

To summarize, U_G is the rotation in the 2-dimensional space spanned by $|\alpha\rangle$ and $|\beta\rangle$. In particular, repeated applications of the Grover operator rotate the initial state vector $|s\rangle$ close to $|\beta\rangle$. For applications of U_G in the order of $O(\sqrt{N})$, a measurement of the state in the computational basis produces with high probability $|\beta\rangle$ as the outcome, that is, the solution to the search problem.

1.4. Search by quantum walks

The search problem first introduced by Grover can be reformulated in terms of search based on a continuous-time quantum walk on a graph [3].

In order to do so we need to modify the quantum walk Hamiltonian of eq. (1.12) so that the vertex $|w\rangle$ is special. Therefore, an *oracle Hamiltonian* is introduced:

$$H_w = -|w\rangle\langle w| \quad (1.18)$$

that has energy zero for all but the vertex $|w\rangle$ for which it has energy -1 . Therefore the Grover problem becomes finding the ground state of this Hamiltonian.

To implement the search we consider the Hamiltonian

$$H = \gamma L + H_w = \gamma L - |w\rangle\langle w| \quad (1.19)$$

where L is the Laplacian of the graph G , that as we have seen in Section 1.2 governs the evolution of the quantum walk.

The quantum search works as following:

- we consider the balanced superposition of all possible states, namely

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_j |j\rangle \quad (1.20)$$

- we run the quantum walk for a time T and find the corresponding evolved state using the Hamiltonian H

$$|\psi(T)\rangle = U(T)|s\rangle = \exp\left\{-iHT\right\}|s\rangle \quad (1.21)$$

(Note that this evolution is valid only for time-independent Hamiltonians.)

- we then measure the occupation probability of the target node

$$p = |\langle w|\psi(T)\rangle|^2. \quad (1.22)$$

The objective is to find the optimal value of γ so that the success probability $|\langle w|\psi(T)\rangle|^2$ is as close as possible to 1 for the smallest T .

It is interesting to notice that the oracle introduced in this fashion is exactly the continuous-time version of the reflection R_w in Grover's algorithm [11] because by itself it only evolves the marked vertex $|w\rangle$ by a phase

$$e^{-i|w\rangle\langle w|t}|w\rangle = e^{-it}|w\rangle \quad (1.23)$$

while leaving the other vertices unchanged, thus making this the continuous-time version of a yes/no oracle.

We can turn our attention on trying to understand why we should expect this algorithm to give a success probability for some values of γ, T . In order to do so we frame the problem in terms of Hamiltonian spectrum [3]. In particular we look at the two extremes, namely $\gamma \rightarrow \infty$ and $\gamma \rightarrow 0$.

- as $\gamma \rightarrow \infty$ the contribution of H_w is negligible, to the point that the ground state of H is $|s\rangle$ ⁵
- on the other hand, if $\gamma \rightarrow 0$ the contribution of the Laplacian to the overall Hamiltonian H disappears and thus the ground state of H is close to $|w\rangle$

For some intermediate γ we have that the Hamiltonian will drive transitions between the two states, thus rotating the state from $|s\rangle$ to one with substantial overlap with $|w\rangle$. In particular this transition will happen at a time that depends on the Hamiltonian eigenvalues separation between the first and ground state, namely $1/(E_1 - E_0)$.

1.4.1 Search on the complete graph

We now look at the complete graph with N nodes and represents the simplest example of the quantum walk search application [3].

We begin by noticing that adding a multiple of the identity matrix to the Laplacian only contributes a global unobservable phase. We add $-NI$ so that:

$$L - NI = N|s\rangle\langle s| \quad (1.24)$$

This gives us the following Hamiltonian:

$$H = -\gamma N|s\rangle\langle s| - |w\rangle\langle w| \quad (1.25)$$

In particular we know that this Hamiltonian acts non-trivially in a two dimensional subspace spanned by $|s\rangle$ and $|w\rangle$, thus making it straightforward to compute its spectrum. If we use the $\{|s\rangle, |w\rangle\}$ basis, the Hamiltonian can be expressed in the following way:

$$H = \frac{-1}{N} \begin{pmatrix} N+1 & \sqrt{N-1} \\ \sqrt{N-1} & N-1 \end{pmatrix} \quad (1.26)$$

where γ is set to $1/N$. Applying the time-evolution operator to the initial state $|s\rangle$, the system at time t will be

$$|\psi(t)\rangle = e^{it} \begin{pmatrix} \frac{1}{\sqrt{N}} \cos\left(\frac{t}{\sqrt{N}}\right) + i \sin\left(\frac{t}{\sqrt{N}}\right) \\ \sqrt{\frac{N-1}{N}} \cos\left(\frac{t}{\sqrt{N}}\right) \end{pmatrix} \quad (1.27)$$

⁵Note that regardless of the graph topology considered, $|s\rangle$ is the ground state of the Laplacian, with eigenvalue 0.

and for $t = \pi\sqrt{N}/2$ the system reaches a success probability of 1, namely

$$p = \left| \langle w | \psi(t) \rangle \right|_{t=\pi\sqrt{N}/2}^2 = 1 \quad (1.28)$$

Thus the walk rotates the state from $|s\rangle$ to $|w\rangle$ in $O(\sqrt{N})$.

1.5. Search by adiabatic evolution

We now address the computation by adiabatic evolution firstly introduced by Farhi et al., that takes advantage of the adiabatic theorem to find the solution of a computational problem [4]. We begin by looking at the *adiabatic theorem* and its implication. Then, we overview the adiabatic implementation of the algorithm for solving the unstructured search problem which has a time scaling of $O(N)$. Lastly, we see how applying the adiabatic theorem *locally* can find the solution in a time of order $O(\sqrt{N})$ which is optimal.

1.5.1 Adiabatic theorem

A quantum system evolves according to the Schroedinger equation

$$i \frac{d}{dt} |\psi(t)\rangle = H(t) |\psi(t)\rangle. \quad (1.29)$$

The instantaneous eigenstates and eigenvalues of $H(t)$ are

$$H(t) |\psi_l(t)\rangle = E_l(t) |\psi_l(t)\rangle \quad (1.30)$$

such that $E_0(t) \leq E_1(t) \leq \dots \leq E_{N-1}(t)$.

The adiabatic theorem states that if the gap between the two lowest energy levels, $E_1(t) - E_0(t) > 0$, is strictly greater than zero then for $T \rightarrow \infty$ the probability of being in the ground state is equal to one, namely

$$\lim_{T \rightarrow \infty} \langle \psi_0(T) | \psi(T) \rangle = 1 \quad (1.31)$$

This means that if the system is chosen to evolve at a slow enough rate, the instantaneous Hamiltonian will remain in the ground state throughout the evolution. Furthermore it is convenient to parametrize the Hamiltonian as $H(s)$, where $s = t/T$, with $t \in [0, T]$ so that $s \in [0, 1]$. Let us now define the minimum

energy gap between the lowest eigenvalues of H by

$$g_{min} = \min_{0 \leq s \leq 1} (E_1(s) - E_0(s)) \quad (1.32)$$

This allows to find a minimum time T^* such that, for $T \gg T^*$ the probability of being in the ground state $|\psi_0\rangle$ is arbitrarily close to 1. In details, T^* is given by:

$$T^* = \frac{\varepsilon}{g_{min}^2} \quad (1.33)$$

where

$$\varepsilon = \max_{0 \leq s \leq 1} \left| \left\langle \psi_1(s) \left| \frac{dH(s)}{ds} \right| \psi_0(s) \right\rangle \right|. \quad (1.34)$$

For future convenience we will refer to the matrix element as $\langle \frac{dH}{ds} \rangle$, with $s \in [0, 1]$.

Let us now discuss how to take advantage of the adiabatic theorem. Let us consider a problem Hamiltonian H_P whose ground state is not so straight forward to find; on the other hand we can prepare the system in a beginning Hamiltonian H_B whose ground state is known. The problem Hamiltonian encodes the solution of the problem, while the beginning Hamiltonian is a tool to easily prepare the state to be evolved. The adiabatic evolution then consists, assuming that the ground state of H_P is unique, in having a time dependent Hamiltonian $H(s)$ that interpolates between H_B and H_P

$$H(s) = (1 - s)H_B + sH_P \quad (1.35)$$

In this way, by tuning the value of s , the Hamiltonian changes from H_B to H_P . Therefore, if the system evolves sufficiently slowly it reaches the ground state of the problem Hamiltonian, that is exactly the solution.

1.5.2 Global adiabatic evolution

Let us now apply the adiabatic theorem to the unsorted search problem [10]. As done in Section 1.3 we consider an unsorted database of N elements such that $N = 2^n$, so that the elements in this particular basis can be written as $|i\rangle$, with $i = 0, \dots, N - 1$. The marked state can be then denoted as $|w\rangle$, and the initial

state is the superposition of all the elements $|i\rangle$:

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{i=1}^{N-1} |i\rangle \quad (1.36)$$

With this in mind we design two particular Hamiltonians H_0 and H_w such that $|\psi_0\rangle$ is the ground state of the first while $|w\rangle$ is the ground state of the latter:

$$H_0 = I - |\psi_0\rangle\langle\psi_0| \quad (1.37)$$

$$H_w = I - |w\rangle\langle w| \quad (1.38)$$

The time-dependent Hamiltonian follows from the adiabatic implementation discussed in Section 1.5, where H_0 is the beginning Hamiltonian and H_w is the problem Hamiltonian. It thus consists in a linear interpolation between H_0 and H_w :

$$H(t) = (1 - s)H_0 + sH_w \quad (1.39)$$

where $s = t/T$ is the linear interpolating schedule.

The search routine runs as usual, with the system prepared in the state $|\psi(0)\rangle = |\psi_0\rangle$ and then we apply the Hamiltonian $H(t)$ for a time T . We are interested in finding the time-dependent condition such that the system evolves sufficiently slowly, allowing us to find the solution $|w\rangle$ with high probability. First, we determine the quantity $\langle \frac{dH}{dt} \rangle$:

$$\langle \frac{dH(t)}{dt} \rangle = \frac{ds}{dt} \langle \frac{dH(s)}{ds} \rangle = \frac{1}{T} \langle \frac{dH(s)}{ds} \rangle \quad (1.40)$$

We then find the eigenvalues of the Hamiltonian and determine the separation g between the ground state and first excited one, namely E_1 and E_0 , as a function of the interpolating schedule s :

$$g = \sqrt{1 - 4 \frac{N-1}{N} s(1-s)}. \quad (1.41)$$

We are interested in the minimum gap g_{\min} which is found for $s = 1/2$. Additionally we find that $|\langle \frac{dH}{ds} \rangle_{0,1}| \leq 1$, therefore Equation (1.33) becomes:

$$\frac{|\langle \frac{dH}{ds} \rangle_{0,1}|}{g_{\min}^2} \leq \varepsilon \quad (1.42)$$

so that the adiabatic condition is verified, and thus the Hamiltonian stays in the ground state at all times, provided that :

$$T \geq N/\varepsilon. \quad (1.43)$$

Therefore the computation time is of order N , showing no speed up compared to the classical search.

1.5.3 Local adiabatic evolution

Roland and Cerf showed that the adiabatic evolution can be improved by applying the adiabatic condition of eq. (1.42) locally instead of globally [10]. If we look at the plot of the separation g we see that the adiabatic condition is critical only for $s = 1/2$ where it reaches its minimum. At the beginning and at the end of

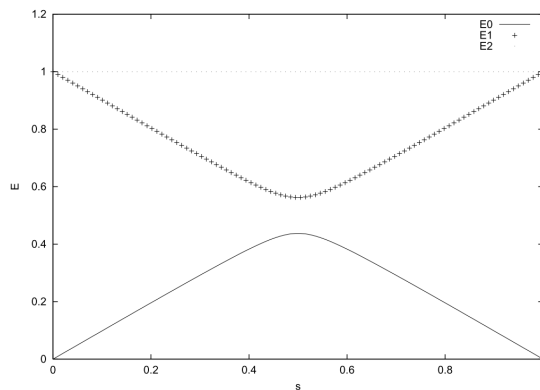


Figure 1.4: Eigenvalue separation of the time-dependent Hamiltonian $H(s)$ as a function of the reduced time s , for $N=64$. *Figure from Roland and Cerf [10]*

the evolution the separation is large enough so that the adiabatic condition and thus the time necessary for the system to evolve adiabatically is small, allowing for better time scaling. The improvement comes in the form of the interpolating schedule, following the idea that it should be steeper for large separation g and flatter for small separation around $s = 1/2$. Let us see how can this be done.

We divide the interval $[0, T]$ into infinitesimal intervals dt and adapt the evolution rate $\frac{ds}{dt}$ to the local adiabatic condition. In this way we can find the optimal $s(t)$, with boundary conditions $s(0) = 0$ and $s(T) = 1$. We find the new adiabatic condition for all time t :

$$\left| \frac{ds}{dt} \right| \leq \varepsilon \frac{g^2(t)}{\left| \left\langle \frac{dH}{ds} \right\rangle_{0,1} \right|} \quad (1.44)$$

Using eq. (1.41) and $|\langle \frac{dH}{ds} \rangle_{0,1}| \leq 1$ and Hamiltonian is chosen to evolve with the interpolating schedule that is solution of the following differential equation (with $\varepsilon \ll 1$):

$$\frac{ds}{dt} = \varepsilon g^2(t) = \varepsilon \left[1 - 4 \frac{N-1}{N} s(1-s) \right] \quad (1.45)$$

After integration we get the following

$$t = \frac{1}{2\varepsilon} \frac{N}{\sqrt{N-1}} \left[\arctan(\sqrt{N-1}(2s-1)) + \arctan \sqrt{N-1} \right] \quad (1.46)$$

By inverting we find the interpolating schedule $s(t)$ as can be seen in fig. 1.5. In order to determine the computation time of this algorithm. To do so we evaluate $s = 1$ and in the approximation that $N \gg 1$ we get:

$$T = \frac{\pi}{2\varepsilon} \sqrt{N} \quad (1.47)$$

which represents a great improvement on Equation (1.43). Indeed we have a quadratic speed up compared to the global adiabatic evolution, and thus this algorithm can be seen as the adiabatic implementation of the Grover's search algorithm.

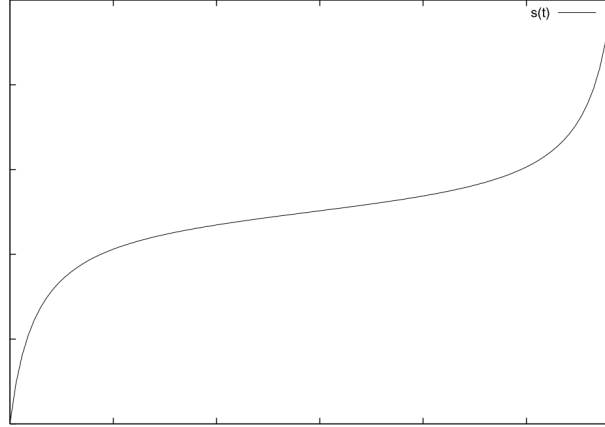


Figure 1.5: Roland and Cerf interpolating schedule for the unstructured search with $N=64$. *Figure from Roland and Cerf [10]*

1.6. Impossibility of a search algorithm based on adiabatic quantum walks

In this section we show that an *adiabatic quantum walks* (AQW) based search algorithm cannot be implemented with the usual Grover's oracle and requires a more sophisticated structure. This section, based on a paper by Wong et al. [11], sets the basis on which our work builds upon.

In the previous sections we discussed Grover's original discrete-time search algorithm, Farhi and Gutmann's quantum walk analogue and Roland and Cerf's local adiabatic analogue, where we analyzed the systems in a 2-dimensional subspace spanned by $\{|\alpha\rangle, |w\rangle\}$.

This allows to visualize the evolution of the quantum algorithm on the two dimensional Bloch sphere and compare the path taken by each different approach. In particular we notice that the original Grover's algorithm and the Roland and Cerf's local adiabatic evolution follow the same path on the xz -plane since they both always have real coefficients, as can be seen in Figures 1.6a and 1.6b respectively. On the other hand the quantum walk formulation of Farhi and Gutmann has an unmeasurable complex phase thus it evolves on a different path on the yz -plane, see Figure 1.6c.

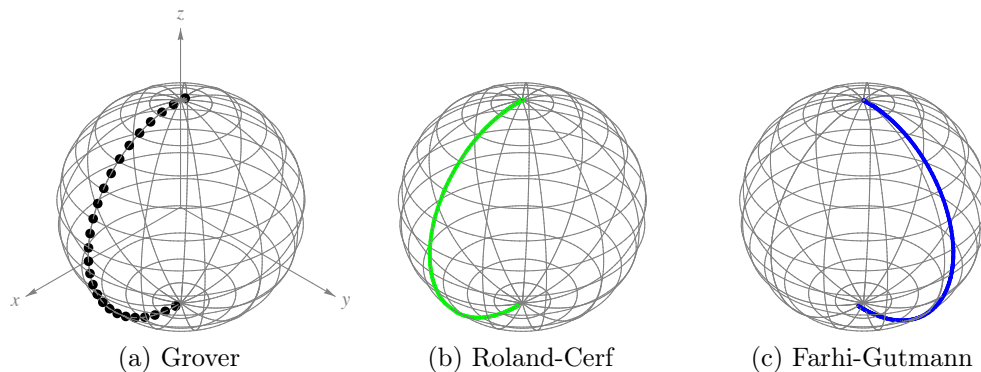


Figure 1.6: **Evolution of the quantum algorithms on the bloch sphere.** The figure shows the evolution of quantum algorithm on the Bloch sphere with marked vertex $|w\rangle$ at the North Pole and the equal superposition of the unmarked vertices $|\alpha\rangle$ at the South Pole, and $N = 1024$: (a) Grover's original discrete-time search algorithm, (b) Roland and Cerf's local adiabatic evolution analogue and (c) Farhi and Gutmann's quantum walk analogue. Wong et al. (2016) [11]

This substantial difference of the path taken by the different algorithms is exactly what Wong et al. investigate, in particular with the goal of determine whether

it is possible to construct an adiabatic algorithm that follows the path of the quantum walks based algorithm.

To do so they construct an Hamiltonian from the following states, with the first represents the ground state and the latter the first (and only) excited state - remember that the Grover's search rotates the initial state over the final state, thus the first excited state is the orthogonal of the ground state.

$$|\psi_0(t)\rangle = \alpha(t)|w\rangle + \beta(t)|\alpha\rangle \quad (1.48)$$

$$|\psi_1(t)\rangle = \beta(t)|w\rangle - \alpha^*(t)|\alpha\rangle \quad (1.49)$$

where the coefficients are given by the components of $|\psi_0(t)\rangle$ of Equation (1.27) ⁶. The Hamiltonian is then given by the linear combination of $|\psi_0(t)\rangle$ and $|\psi_1(t)\rangle$:

$$H(t) = \lambda_0|\psi_0\rangle\langle\psi_0| + \lambda_1|\psi_1\rangle\langle\psi_1| \quad (1.50)$$

which allows to determine the final time-dependent *adiabatic* Hamiltonian - where the time-dependence is given by the interpolating schedule $s(t)$:

$$H(s) = \sqrt[4]{\frac{s(1-s)}{4\varepsilon^2 N}} \left[(1-s)H_0 + sH_f + \sqrt{s(1-s)}H_e \right] \quad (1.51)$$

where the interpolating schedule is given by $s(t) = \sin^2\left(\frac{t}{\sqrt{N}}\right)$. Indeed the system evolves from $t = 0$ to $t = \frac{\pi}{2}\sqrt{N}$ as we would expect. Looking closely, the Hamiltonian H_0 represents the beginning Hamiltonian, H_f the final Hamiltonian - encoding the solution of the search problem - and H_e is an *extra* Hamiltonian, a common technique for manipulating the evolution of the path of adiabatic algorithms⁷. Let us now look at the consequences of this Hamiltonian, looking at H_0 , H_f and H_e individually.

H_0 has ground state $|s\rangle$ and excited state $|s^\perp\rangle$ with respective eigenvalues -1 and $+1$, thus can be written as:

$$H_0 = |s^\perp\rangle\langle s^\perp| - |s\rangle\langle s| \quad (1.52)$$

Similarly, the final Hamiltonian H_f has ground state $|w\rangle$ and excited state $|\alpha\rangle$

⁶Note that the global phase e^{it} is dropped.

⁷The extra Hamiltonian commonly appears as $s(s-1)$, while the one considered in this scenario is given by the square root of such value [11].

with eigenvalues $+1$ and -1 :

$$H_f = |\alpha\rangle\langle\alpha| - |w\rangle\langle w| \quad (1.53)$$

It is immediate to notice that these two Hamiltonians look fairly similar to the initial and final Hamiltonian of the standard adiabatic quantum search algorithm of Equations (1.37) and (1.38). Although the similarities look at first sight promising, the extra Hamiltonian H_e and the interpolating schedule $s(t)$ change completely the evolution of the system. In particular, if we look at the Hamiltonian H_e - given by:

$$H_e = 2i\sqrt{\frac{N-1}{N}}(|\alpha\rangle\langle w| - |w\rangle\langle\alpha|) \quad (1.54)$$

we see that rather than having the standard yes/no oracle given by $|w\rangle\langle w|$, it introduces a more powerful structure, driving the evolution between $|w\rangle$ and $|\alpha\rangle$ instead of just applying a phase to the target state $|w\rangle$ like in the standard Grover's algorithm. This is a confirmation that the oracle is no longer the usual Grover's yes/no oracle, and although this a proof that an adiabatic evolution following the quantum walk path does exist it abandons the usual notion of the oracle and does not solve the search problem itself as originally posed by Grover.

The difference between the Quantum Walks and the Adiabatic Evolution approach is not a measure of the performance of the algorithms, since both, individually, solve the Grover's problem in $O(\sqrt{N})$ time, but illustrate that “*how they compute is different, even though *what* they compute is the same*” [11].

Although the work by Wong et al. shows that is not possible to implement an adiabatic quantum walk algorithm that solves the Grover's search problem, it leaves space to a time-dependent algorithm *inspired* by the adiabatic evolution but that it is free from the constraints of the adiabatic condition. In the next chapter we investigate this possibility, that gives some interesting insights and properties of the quantum walks search algorithm with time-dependent Hamiltonians.