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

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 [10], from the *collision problem* [1] to solving NP-complete problems [6].

The original Grover's algorithm was formulated at 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 [13]. In 1998 Farhi and Gutmann introduced the idea of analog quantum computing, where the system evolves in continous-time following the Schroedinger 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 comes as a natural consequence the idea of comparing its different formulations, highlighting their differences and similarities [13].

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 approach of Farhi and Gutmann, and for completeness we give some results for the complete graph, in which the quantum walk search may be accomplished exactly. 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 considered graphs. 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.

Chapter 1

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 coursive 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 $j = 1, 2, ..., N$.

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]:

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

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. The matrix D is defined as:

$$D = diag(d_1, ..., d_N) \tag{1.2}$$

where d_j is the degree of the vertex j. In this particular context a natural operative basis arises, in which one can associate with 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

$$L = D - A \tag{1.3}$$

where D is the diagonal degrees matrix and A is the adjacenty matrix. The discrete Laplacian operator for the finite, undirected, simple and connected graphs can charactarized 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

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

[2] If 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 consider in this thesis the given definition suffices.

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

As previously mentioned we can associate every vertex with 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 also allows us to write the Laplacian matrix in convenient quantum mechanical form using the projectors $|j\rangle\langle i|$

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

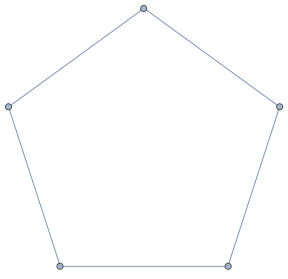
In the following paragraphs we give a brief description of the graphs that will be consider throughout the work.

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

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

A pictorial representation of a cycle graph is given in Figure 1.1 with the corrisponding 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:

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

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

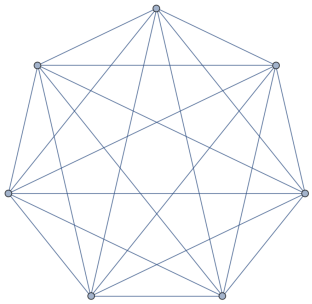


Figure 1.2: Pictorial representation of a complete graph with $N = 7$

1.2. Quantum walks

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

A continous-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 date 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 the probability of transitioning from a node to another, also called *transition rates*. In particular

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

these 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:

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

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

$$T = -\gamma L \tag{1.8}$$

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

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

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

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

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

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

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]:

$$H = -T = L \tag{1.12}$$

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 satisfies 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 mechanicals 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 [10].

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.

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

We can now 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:

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

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

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x}^{N} |x\rangle \tag{1.14}$$

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 $|\beta\rangle$:

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

$$|\beta\rangle = |w\rangle \tag{1.16}$$

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:

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

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.

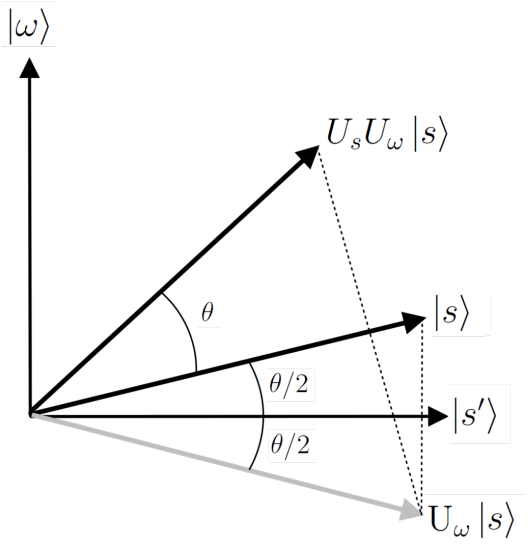


Figure 1.3: **Action of the Grover operator U_G**: The figure shows the action of the Grover operator $U_G = U_s U_w$. Starting from the initial state $|s\rangle$ the action of the Oracle U_w reflects it about the vector $|\alpha\rangle$. Then, the action of U_s is an additional reflection about $|s\rangle$.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

$$H_w = -|w\rangle\langle w| \tag{1.18}$$

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

$$H = \gamma L + H_w = \gamma L - |w\rangle\langle w| \tag{1.19}$$

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

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_j |j\rangle \tag{1.20}$$

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

$$|\psi(T)\rangle = U(T)|s\rangle = \exp\Big\{ - iHT \Big\}|s\rangle \tag{1.21}$$

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

- we then measure the occupation probability of the target node

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

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 continous-time version of the reflection U_w in Grover's algorithm [13] because by itself it only evolves the marked vertex $|w\rangle$ by a phase

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

while leaving the other vertices unchanced, thus making this the continous-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 $\gamma \to \infty$ and $\gamma \to 0$.

- as $\gamma \to \infty$ 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 $\gamma \to 0$ 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 that 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:

$$L - NI = N|s\rangle\langle s| \tag{1.24}$$

This gives us the following Hamiltonian:

$$H = -\gamma N|s\rangle\langle s| - |w\rangle\langle w| \tag{1.25}$$

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 stright-forward to compute its spectrum. If we use the $\{|\alpha\rangle, |w\rangle\}$ basis, the Hamiltonian can be expressed in the following way:

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

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

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

[5]Note 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

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

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

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

The instantaneous eigenstates and eigenvalues of H(t) are given by

$$H(t)|\psi_l(t)\rangle = E_l(t)|\psi_l(t)\rangle \tag{1.30}$$

such that $E_0(t) \leq E_1(t) \leq ... \leq E_{N-1}(t)$.
The adiabatic theorem states that if the gap between the two lowest energy levels, $E_1(t) - E_0(t) > 0$, is stritcly greater than zero, then for $T \to \infty$ the probability of being in the ground state $|\psi_0\rangle$ is equal to one, namely

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

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

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

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:

$$T^* = \frac{\varepsilon}{g_{min}^2} \tag{1.33}$$

where

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

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

$$H(s) = (1 - s)H_B + sH_P \tag{1.35}$$

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 adiabtic theorem to the unsorted search problem [11]. 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 $i = 0, ..., N - 1$. The marked state can be then denoted as $|w\rangle$, and the initial

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

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

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:

$$H_0 = I - |\psi_0\rangle\langle\psi_0| \tag{1.37}$$

$$H_w = I - |w\rangle\langle w| \tag{1.38}$$

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:

$$H(t) = (1-s)H_0 + sH_w \tag{1.39}$$

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 $\langle \frac{dH}{dt} \rangle$:

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

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:

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

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

$$\frac{\left| \langle \frac{dH}{ds} \rangle_{0,1} \right|}{g_{\min}^2} \leq \varepsilon \tag{1.42}$$

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

$$T \geq N/\varepsilon. \tag{1.43}$$

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 [11]. 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

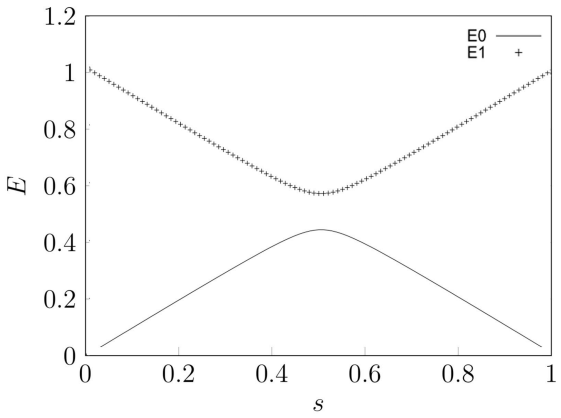


Figure 1.4: Eigenvalue separation of the time-dependent Hamiltonian $H(s)$ as a function of the reduced time s, for N=64. *Figure from Roland and Cerf* [11]

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 adabatic 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:

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

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

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

After integration we get the following

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

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 $N >> 1$ we get:

$$T = \frac{\pi}{2\varepsilon}\sqrt{N} \tag{1.47}$$

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.

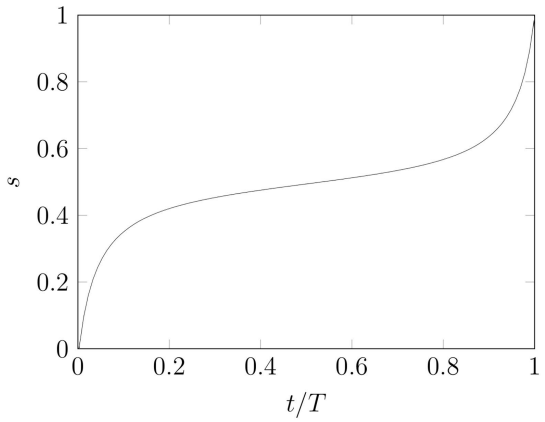
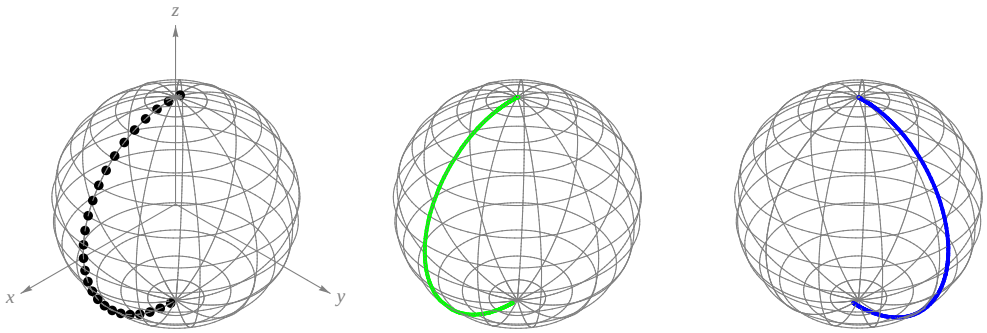


Figure 1.5: Roland and Cerf interpolating schedule for the unstructured search with N=64. *Figure from Roland and Cerf* [11]

1.6. Impossibility of a search algoritm 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. [13], 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.



(a) Grover (b) Roland-Cerf (c) Farhi-Gutmann

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)* [13]

This substantial difference of the path taken by the different algorithms is exacly 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 representing the ground state and the latter the first (and only) excited state – rememeber 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.

$$|\psi_0(t)\rangle = \alpha(t)|w\rangle + \beta(t)|\alpha\rangle \tag{1.48}$$

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

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 $|\psi_1(t)\rangle$:

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

which allows to determine the final time-dependent *adiabatic* Hamiltonian – with the time-dependence given by the interpolating schedule $s(t) = \sin^2\left(\frac{t}{\sqrt{N}}\right)$:

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

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 tecnique for manupulating the evolution of the path of adiabatic algorithms[7]. 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:

$$H_0 = |s^\perp\rangle\langle s^\perp| - |s\rangle\langle s| \tag{1.52}$$

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

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

[7] The extra Hamiltonian commonly appears as $s(s-1)$, while the one considered in this scenario is given by the square root of such value [13].

with eigenvalues $+1$ and -1:

$$H_f = |\alpha\rangle\langle\alpha| - |w\rangle\langle w| \tag{1.53}$$

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:

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

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 targer 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 exists 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 " [13].

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 constrains 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.

Chapter 2

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

The main goal of this thesis is to study quantum walks with time dependent Hamiltonians, focusing in particular on their application to the spatial search problem on graphs. The general idea is trying to improve a time-independent implementation of quantum walks search using a time-dependent Hamiltonian analogous to the one used in the adiabatic evolution. In order to determine whether this new approach produces successfull results we study two selected graph topologies: the cycle graph – for which the time-independent approach is not able to solve the search problem – and the complete graph for which the search problem is solved for both the time-independent and adiabatic evolution approaches.
We compare the two methods for the *search*, *localization* – which represents a search without the needs to optimize the time – and a measure of *robustness*.

2.1. Search with time-dependent Hamiltonian

In Section 1.4 we discussed the use of quantum walks for the spatial search problem [3] and the application to the complete graph where the search problem is solved. We then illustrated how the adiabatic theorem can be used to solve computational problems with an adiabatic evolution of a quantum system [4]. Efforts to combine the two approaches showed that the search problem can be

solved only with structures stronger than the usual Grover's oracle [13], thus making an *Adiabatic-Quantum-Walk-Search* impossible. However this leaves space to a merely time-dependent quantum walk search, that takes advantage of a time-dependent implementation similar to the adiabatic evolution but that is not bounded to the strict adiabatic-theorem conditions and the limitation of the standard Grover's oracle.

2.1.1 Time-dependent quantum walks

Following from the adiabatic evolution discussed in the preliminaries, we consider a search Hamiltonian that interpolates between an initial Hamiltonian, i.e. the Laplacian of the graph L, and the final oracle Hamiltonian $H_w = \gamma|w\rangle\langle w|$

$$H(s) = (1-s)L - s\gamma|w\rangle\langle w| \tag{2.1}$$

where the interpolation schedule $s = s(t)$ goes from 0 to 1 as the time t goes from 0 to the runtime T.

In order to find the evolved ground state of the beginning Hamiltonian we need to determine the evolution operator that for a time-dependent Hamiltonian is given by:

$$S(t, t_0) = \mathrm{T}\ \exp\Big\{ -i \int_{t_0}^{t} dt' H(t') \Big\} \tag{2.2}$$

where T is the time-ordering operator. Since we are only interested in the evolved state, having the exact evolution operator is irrelevant. We therefore proceed by solving the differential Schroedinger equation:

$$i\frac{d}{dt}|\psi(t)\rangle = H|\psi(t)\rangle \tag{2.3}$$

Recalling that we are dealing with matrices and vectors in an N-dimensional Hilbert space, we solve N-differential equations of the form

$$\frac{d}{dt}|\psi_i(t)\rangle = \sum_j H_{ij}|\psi_i(t)\rangle \tag{2.4}$$

with the boundary conditions $|\psi(0)\rangle = |\psi_0\rangle$, where the latter is the ground state of the Laplacian.

2.1.2 Interpolating schedule s(t)

As pointed by Wong the interpolating schedule s(t) plays a crucial role in the evolution of the system and in the overall scaling of the algorithm. The original adiabatic evolution by Farhi and Gutmann [4] uses a linear interpolating schedule defined as $s_L(t) \equiv \frac{t}{T}$. Roland and Cerf show that in order to obtain a quadratic speedup for the complete graph a non linear schedule is essential [11].

Thus, from a linear interpolating schedule we consider also quadratic and cubic schedules:

$$s_S(t) \equiv \sqrt{\frac{t}{T}} \qquad s_C(t) \equiv \sqrt[3]{\frac{t}{T}} \tag{2.5}$$

We then consider the interpolating schedule analitically derived by Roland and Cerf for the unstructured search discussed in Section 1.5.3. As we have already mentioned the shape of this interpolating schedule follows from gap $g(s)$ between the lowest two eigenvalues, changing faster when the gap is large, while it evolves slower when the gap is small. We therefore take these key aspects of $s(t)$ – derived for the unstructured search – and consider a similar non-linear interpolating schedule defined as follows

$$s_{NL}(t) = \frac{1}{2}\left[\left(2\frac{t}{T} - 1\right)^3 + 1\right] \tag{2.6}$$

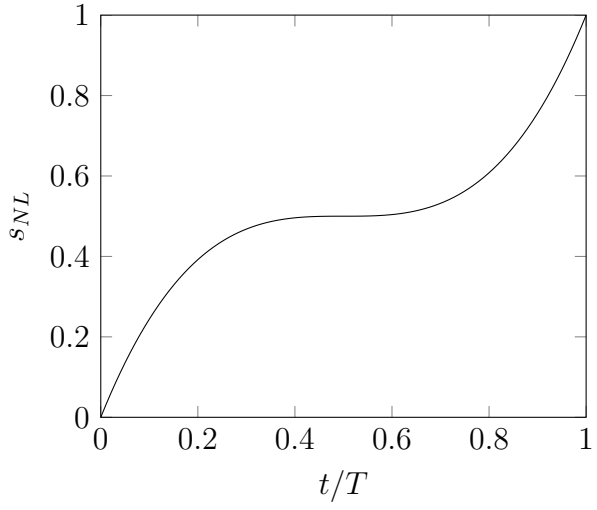


Figure 2.1: **Non-linear interpolating schedule $s_{NL}(t/T)$.** The figure shows the non-linear interpolating schedule s_{NL} that we introduced to improve the standar linear one of Farhi and Gutmann [4]. It draws the key aspects from the interpolating schedule derived by Roland and Cerf [11], namely it evolves faster for large energy separation g and slower for small separation.

2.1.3 Multiple runs for one search

To give a complete picture of the usefulness of the time-depedent approach, we consider the possibility of repeating the search multiple times. If the probability at which the solution is found is $p = 1$ the search is *perfect* and the problem is solved. However, if the proabability is less than one, i.e. *imperfect search*, the problem can be solved by searching multiple times . Since the results of the search are checked independently, a single successfull search is sufficient, and this kind of routine is efficient as long as the probability $\left|\langle\psi(T)|w\rangle\right|^2$ is greater than $1/poly(N)$ (where N is the dimension of the graph) [8] – which as we shall later see is verified for all the scenarios considered.

Repeating the search multiple times does however come at a cost. It is indeed necessary to take into account for a non-zero *initialization* time t_{init} to prepare the system in the correct state as well a physical time for the measurement. Therefore, computing multiple searches with small T becomes less efficient than less searches with larger T, where the quantity t_{init} makes a lesser contribution to the overall T.

2.2. Selected topologies: cycle and complete graph

Throughout our analysis we will focus on two selected graph topology, the cycle graph $Cy(N)$ and the complete graph $C(N)$.

As previously discussed in Section 1.4.1 and **??** the **complete graph** represents the best case scenario since it has been shown to solve the search problem both for the standard time-independent quantum walk approach [3] and the local adiabatic evolution [11] with a quadratic speed up. An adiabatic implementation of the quantum walk search does not work with the usual Grover's oracle, requiring a more elaborate structure [13], however as we shall later see a merely time-dependent approach might give interesting results in terms of robustness and localization.

The **cycle graph** on the other hand is not able to solve the search problem with the time-independent approach, and can give some interesting insights on the performance of the time-dependent quantum walk search.

The goal is therefore to study the time-dependent approach on a non-working topology – the cycle graph – and for completeness show its application to the already perfect search on the complete graph.

2.3. Characterization of the results: Search, Localization and Robustness

Firstly it is necessary to define what type of results we are looking for. We begin by showing the difference between **search** and **localization**. Then we introduce a measure for the **robustness** of the search algorithm.

2.3.1 Search vs Localization

In order to study the performance of the search algorithms we have to characterize two particular classes of results, the (optimized) **search** and the **localization**, that help us determine whether the time-dependent approach brings any advantages.

- the (optimized) **search** describes the usual search, namely the finding of the solution with high probability (possibly unitary) for the smallest time as possible. As previously mentioned we also take into account the possibility of repeating the search multiple times.

- we call **localization** the finding of the solution with high probablity without the need to optimize the time. This description becomes necessary if we take into account the adiabatic nature of the time-dependent approach, that guarantees unitary probability for large T.

2.3.2 Robustness

We have seen that the probability of a search problem depends on the time T at which the quantum state is measured and the parameter γ. The optimal highest probability clearly is given by the optimal combination of T and γ. These two parameters might be affected by noise or perturbation, leading to variation from the maximum probability. Therefore it is interesting to define a quantity that is able to quantify this phenomenon.

Following S. Hung *et. al* [12] we define the **robustness**, a quantitative measure representing the variation on the probability due to some perturbation/noise on γ. We begin by finding the highest probability p, evaluated with the single or multiple run for one search approach. For the corresponding (T, γ) combination we evaluate the robustness as follows:

$$R^{\pm} = p(T, \gamma) - p(T, \gamma \pm \delta) \tag{2.7}$$

where δ is some positive perturbation of the γ parameter. As we shall later see this quantity is given in terms of some percentage of γ. To find a unique value for the robustness an average of $R^{\pm}$ is done:

$$R_{\gamma} = \left[\frac{R^{+} + R^{-}}{2}\right] \tag{2.8}$$

The quantity R should be positive[1], since the (T, γ) combination corresponds to the hightest probability. Additionally we also notice that the perturbation on the parameter has equal probability of being positive or negative, thus the average

[1] Although not considered in our work, it is also possible to evaluate the robustness for non-maximal probability. In that particular scenario the value of the robustness could be negative.

ponderates between these two possibilities.

As we mentioned, the variation in probability could also be due to some error on the time T at which the state is measured. We can extend the measure of robustness to this particular scenario, where the value is similarly evaluated:

$$R^{\pm} = p(T, \gamma) - p(T \pm \tilde{\delta}, \gamma) \tag{2.9}$$

and R follows directly from Equation (2.8). In this scenario the robustness in respect to T is not necessarily positive, in particular for the time-dependent approach, since – given the adiabatic nature of the algorithm – the probability increases with time. In order to differentiate between these two measures of robustness we call them **γ-robustness** and **T-robustness** respectively.

Although the value of R does not have any absolute physical significance, it fits well for our specific scenario where the interest is focused on the comparison between two specific approaches. Therefore this quantity will be used to characterize a particular approach as **more** or **less robust**, where the first is characterized by a smaller value of R compared to the latter.

2.4. Results for the cycle graph

We now address the results for the cycle graph. We begin by evaluating the probabilities for the time-independent and the time-dependent Hamiltonians. We compare the localization of the two approaches, followed by the search - which requires the introduction of a new quantity that takes into account the possibility of doing multiple runs for one search. Lastly we study the robustness of the time-dependent approach and determine whether the newly introduced Hamiltonian makes the search more robust than the time-independent one.

2.4.1 Time-independent benchmarks

The first step of the analysis is to compute time-independent benchmarks. We comput the probability over a (T, γ) grid, with $T \in [0, N]$ and $\gamma \in [0, 2.5]$, where N is the dimension of the graph considered. An initial run shows that the probability does not increase with time (Figure 2.7), thus the need to evaluate for T greater than $T = N$ proves to be unnecessary. In addition, Grover's algorithm for the unstructured search has a time scaling of $O(\sqrt{N})$ and the Farhi and Gutmann's global adiabatic evolution is $O(N)$, thus focusing on $T \leq N$ suffices.

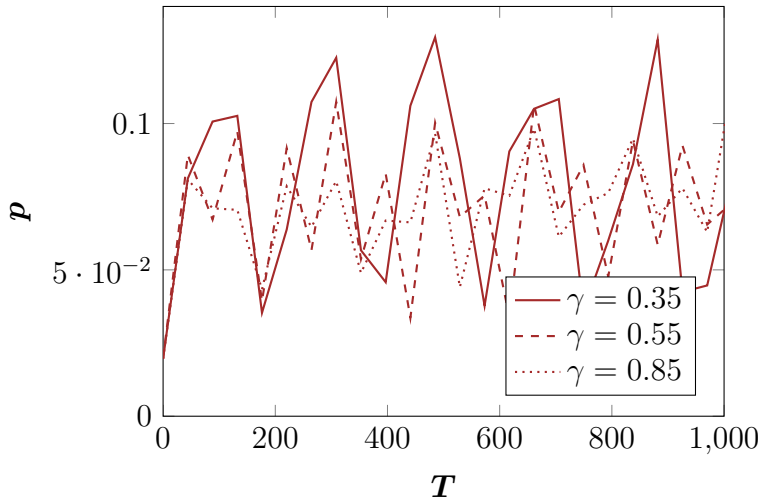


Figure 2.2: **Probability for a $Cy(51)$ with sampled γ**: The figure shows the probability for the cycle graph $Cy(51)$, evaluated with the time-independent Hamiltonian and a few sampled γ. We can see that the probability does not increase significantly with time, therefore we can focus our analysis on $T \leq N$.

Throughout the analysis we will consider graphs up to $N = 71$, with only odd dimensions which makes a easier oracle placement in the center vertex of the graph[2].

To display the results in an intuitive way we used a heatmap plot, which gives a good idea on the probability for varying combinations of (T, γ):

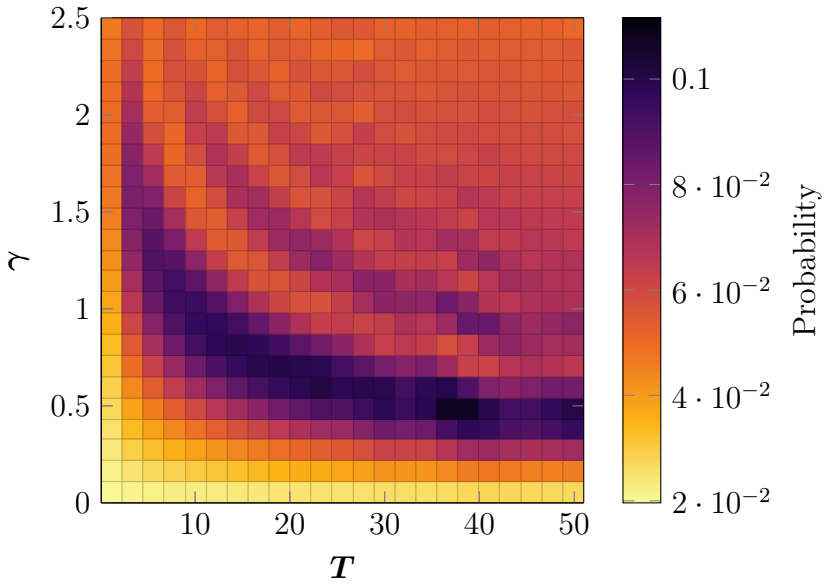


Figure 2.3: **Probability distribution for the time-independent Hamiltonian**. The figure shows the probability distribution for the cycle graph $Cy(51)$ evaluated using the time-independent Hamiltonian, providing the necessary benchmarks. Note that dark color regions do not represent probabilities close to one.

Comments on the probability distribution[3]

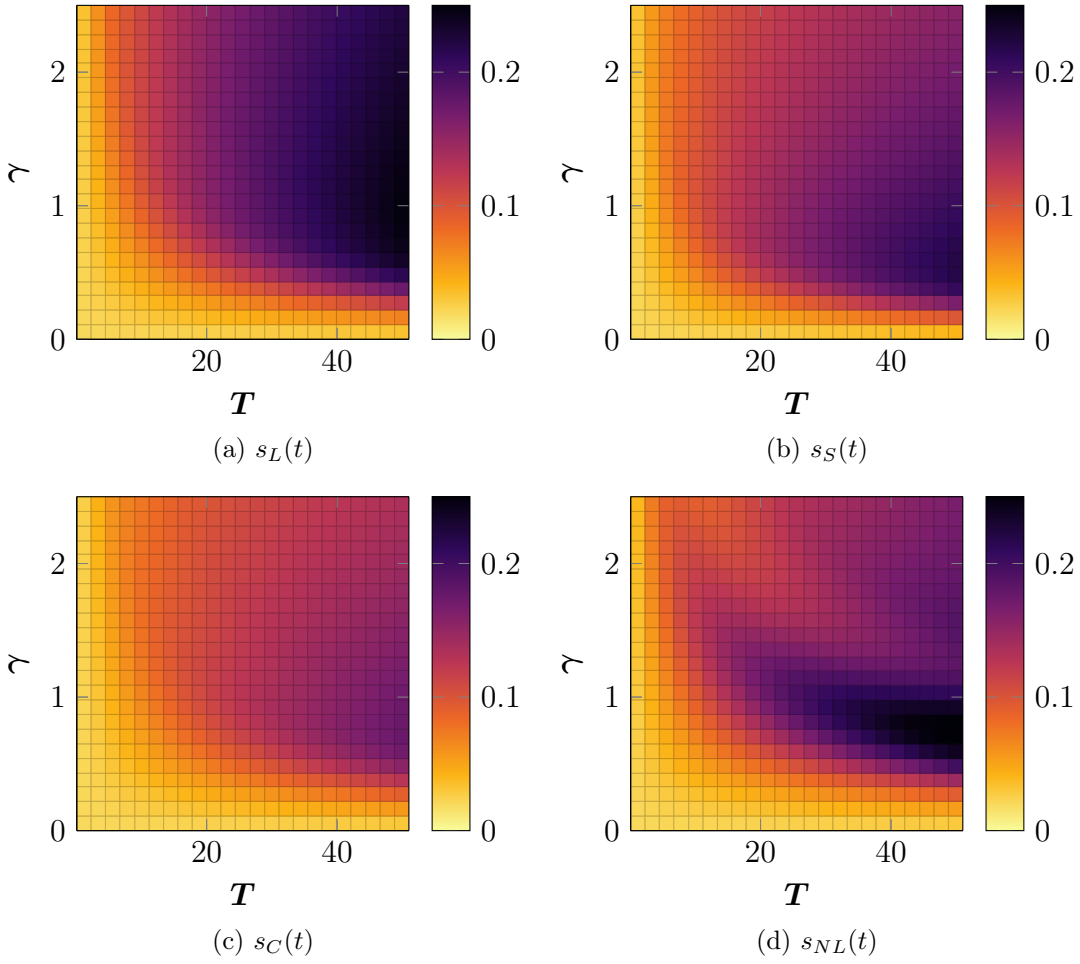
In Figure 2.3 we see that the probability does not increase smoothly with increasing time, but shows peaks (dark regions) and valleys (lighter regions). Additionally high probability regions are scattered throughout the grid and do not appear necessarily for large T. We will later see that this is a weakness of the time-independent approach, since a small variation of the parameter γ leads to possibly great variation of the probability.

[2]The position of the oracle is irrelevant for the graph topologies considered, since every node is equivalent to all the $N - 1$ others. Nevertheless odd dimensions allows to place the oracle in the central vertex of the graph, namely $|\frac{N+1}{2}\rangle$.

[3]The use of *probability distribution* is not to be intended for the meaning it has in probability theory. In our scenario it describes the distribution of the values of probability for the combinations of the variables γ and T

2.4.2 Time-dependent results

Similarly we compute the probability with the time-dependent Hamiltonian using the interpolating schedules introduced in Section 2.1.2. To easily compare the two methods we consider the same time $T = N$ and γ used for the time-independent benchmarks. Indeed, from an initial run we see that the γ parameter affects the probability similarly to the time-independent approach, namely the probability tends to be higher for smaller values of γ.
The probability is then evaluated for all the interpolating schedules $s(t)$, and the results are again presented with an intuitive heatmap plot.



(a) $s_L(t)$ (b) $s_S(t)$

(c) $s_C(t)$ (d) $s_{NL}(t)$

Figure 2.4: **Probability heatmap plot for the time-dependent Hamiltonian, for different shapes of s(t) and $Cy(51)$.** The figure shows the probability heatmap plots for a circular graph of $N = 51$ evaluated using the time-dependent Hamiltonian using the following interpolating schedules: (a) linear, (b) $s_S(t)$, (c) $s_C(t)$ and (d) non-linear $s_{NL}(t)$.

Comments on the probability distribution

Compared to the time-independent Hamiltonian approach we can clearly see that the probability distribution is smoother - has no valley and peaks - for both constant γ and T (any orizontal and vertical sections, respectively). In particular we notice that the probability increases for increasing time, and as we shall later see, for large enough T it reaches probability equal to one.
If we look at the different interpolating schedules it is immediately evident that $s_S(t)$ and $s_C(t)$ schedules perform poorly compared to the linear $s_L(t)$. On the other hand $s_{NL}(t)$ schedule performs similarly to $s_L(t)$, but it has a significantly different probability distribution that might affect its robustness.

2.4.3 Comparison: localization

We now compare the localization properties of the two algorithms. As we have already mentioned, from an intial look at the probability heatmap plots we discovered the following:

- **Time-Independent QW**: the time-independent based algorithm is unable to solve the search problem with a single iteration, making it necessary to run multiple searches. This implies that the approach does not show any localization properties, in fact the probability does not increase with time as seen in Figure 2.3.
- **Time-Dependent QW**: the time-dependent based search on the other hand solves the search problem with a single iteration, although that happens for large values of T, as can be seen in Figure 2.5. This is indeed a consequence of the *adiabatic inspired* implementation, for which at large T the probability goes to one.

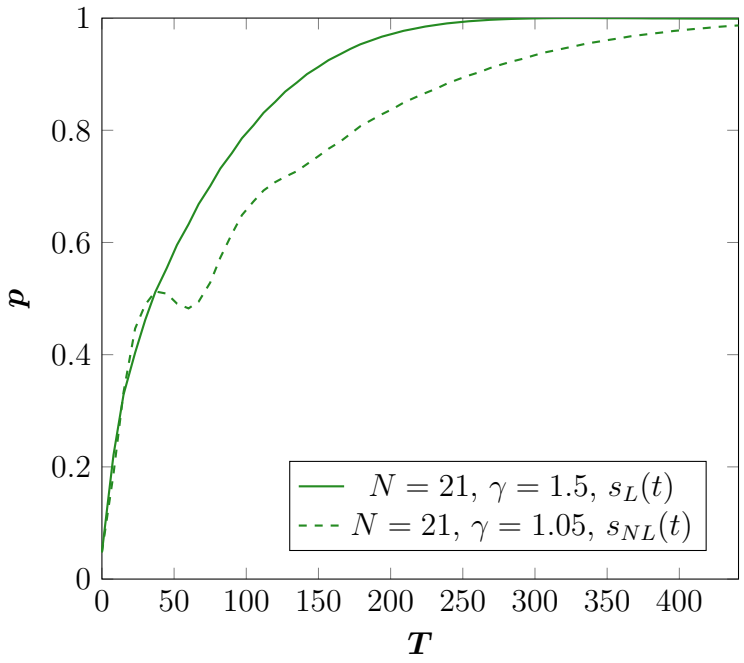


Figure 2.5: **Probability for a $Cy(21)$ with sampled γ up to large T.**: The figure shows the probability distribution for the cycle graph $Cy(21)$, evaluated with the time-dependent Hamiltonian using linear s_L (solid) and non-linear s_{NL} (dashed) interpolating schedules. We can see that the probability increases with time, as expected.

Although the time-dependent approach is able to get to unitary probability for large T, it is able to produce large enough probabilities in much less time, as we can see from this plot. This is a consequence of the fact that the probability does not grow linearly with time, thus needing larger T closer it gets to $p = 1$. If we look at the following plot - evaluated for a $Cy(51)$ and linear $s_L(t)$ - we can see that the algorithm solves the search problem with $p > 0.99$ in a time $T \approx 1200$, while it is able to solve it with $p > 0.8$ in $T/2 \approx 600$.

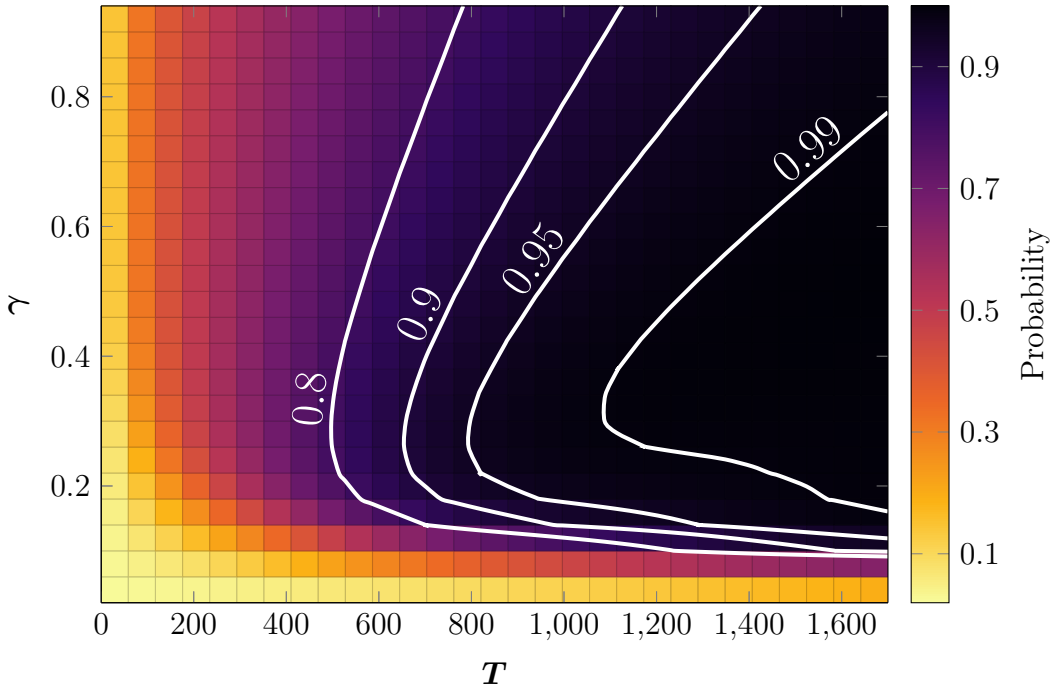


Figure 2.6: **Localization at large T for the time-dependent Hamiltonian**. The figure shows the probability heatmap for the time-dependent Hamiltonian using the linear interpolating schedule $s_L(t)$ up to large values of T. Although, as we can see from the white contour lines, the algorithm gets to $p > 0.99$ for a large value of T, it is able to produce $p = 0.9$ in $T/2$ and $p = 0.8$ in $T/3$.

2.4.4 Comparison: search

In order to compare the two approaches for the search it is clear that we cannot simply consider the time at which the solution is found with unitary probability, since that particular T does not exist for the time-independent approach and is not optimized for the time-dependent one as seen for the localization results. Therefore, as previously mentioned, we consider the possibility of doing multiple runs for one search. For this reason we introduce the following quantity

$$\tau = \min \left(\frac{T}{p} \right)_{T,\gamma} \tag{2.10}$$

where the minimization is done over T and γ [4]. The quantity $1/p$ represents the number of runs necessary to get to unitary probability (statistically), and combining it with T gives the total time necessary get to $p = 1$. Minimizing over the combination of T and p gives the smallest time necessary to solve the search problem with unitary probability using the multiple runs approach.
The minimization thus consists in finding for fixed T the maximum probability, evaluating then the quantity T/p and finding the minimum.

Additionally, the number of runs performed - from now on referred to as **iterations** - given by

$$I = \min \left(p^{-1} \right)_{T,\gamma} \tag{2.11}$$

might give some useful insights on the performance of the approach, in particular if you take into account the initialization time t_{init}, since large number of I are penalized by such t_{init}.

However, this approach poses a few problems. In fact if we look at how the quantity T/p varies for varying T, we discover that the minimum of such quantity always corresponds to the smallest T available, regardless of the type of Hamiltonian and interpolating schedule $s(t)$ considered. The following plot indeed shows, for a sampled γ, the shape of T/p with increasing time.

[4] Remember that the probability depends directly on the combination of T and γ.

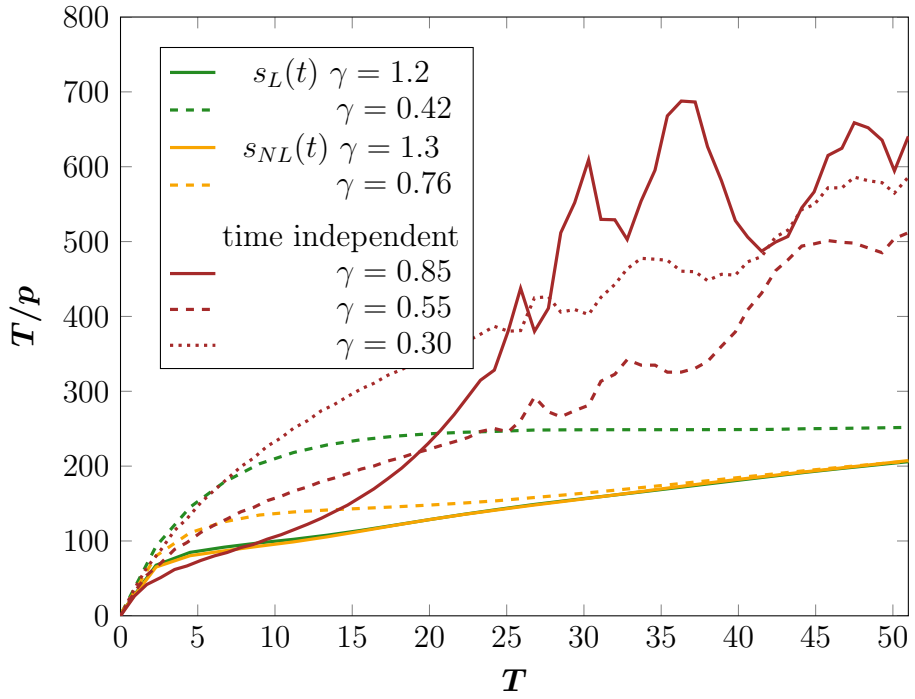


Figure 2.7: ***T/p* for sampled γ, showing that τ always corresponds to the smallest *T***. The plot shows T/p for some sampled values of the γ parameter, using the linear s_L (green) and non-linear s_{NL} (orange) interpolating schedules for the time-dependent Hamiltonian, and the time-independent Hamiltonian (red). It is clear that $\tau = \min(T/p)$ will always correspond to the smallest T available, regardless of the interpolating schedule and the type of Hamiltonian, requiring therefore a constrain on the time.

In addition, in Section 2.1.3 we mentioned that when dealing with multiple runs we need to take into account an initialization time t_{init}. It becomes therefore necessary to introduce a minimum value of time, since for small T the contribution given by t_{init} can have a significant impact. We call this particular T **lower bound time**, referred to as $T_{\min}$, and study its effect on τ and the overall performance of the time-independent and time-dependent algorithms.

τ and I for increasing lower bound time $T_{\min}$

We begin by studying how τ and I vary with increasing lower bound time $T_{\min}$. In the following plot we show the shape of τ with the time-independent approach and the time-dependent one; since we are interested in the general effect of $T_{\min}$ on τ and I we only consider the linear interpolating schedule $s_L(t)$[5]. For sake of

[5]We have seen in Section 2.4.2 the probability distribution for the different interpolating schedules is quite similar, therefore looking only at $s_L(t)$ is sufficient.

semplicity we study the cycle graph $Cy(51)$ with time up to $T = N$

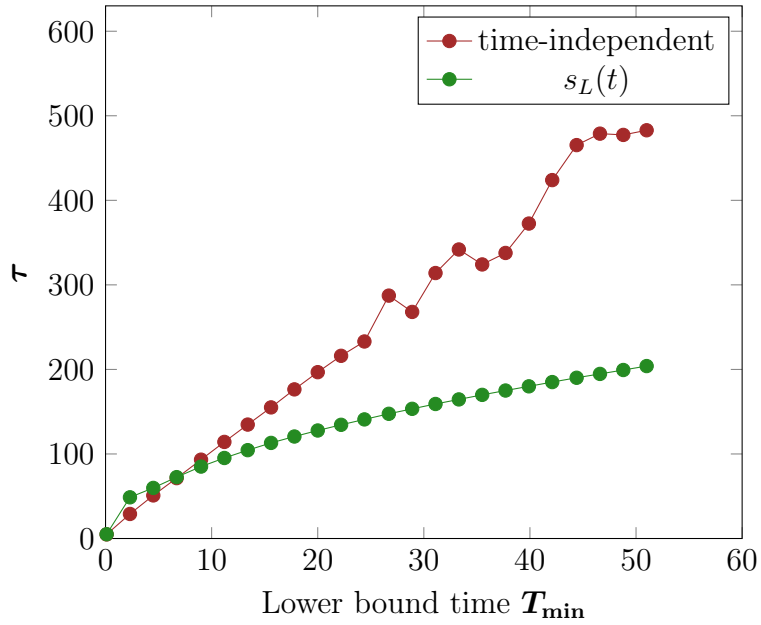


Figure 2.8: **τ for increasing lower bound time $T_{\min}$, $Cy(51)$.** The plot shows τ for increasing values of lower bound time $T_{\min}$, using the time-independent Hamiltonian (red) and time-dependent Hamiltonian (green) with $s_L(t)$ and evaluated for a $Cy(51)$. We see that for times smaller than a characteristic time T^* the time-independent approach performs slightly better, while for large time the time-dependent one performs significantly better.

As we can see from the plot, the distributions can be divided into two sections marked by a particular T^*, representing the time at which the two distributions - time-dependendent and time-independent - intersect (for the time being the value of such time is not of our interest):

- for $T_{\min} < T^*$ the time-dependent approach has a comparable performance with the time-independent one, although the latter has a slight advantage.
- for $T_{\min} > T^*$ however the time-dependent approach performs significantly better, in particular with increasing $T_{\min}$

The behaviour for large T is to be expected, considering that the time-dependent approach shows localization properties and the probability increases with increasing time as we showed in Figure 2.6 and in Section 2.4.3, in contrast with the time-independent approach that does not show localization properties.
What this shows is that the choice of $T_{\min}$ has a great impact on the outcome

of our time-dependent approach, making it a successfull or unsuccesfull alternative.

Let us now look at the distribution of the number of iterations I. The following plots shows I for increasing $T_{\min}$ up to N, using the time-independent Hamiltonian and time-dependent one with the linear interpolating schedule $s_L(t)$.

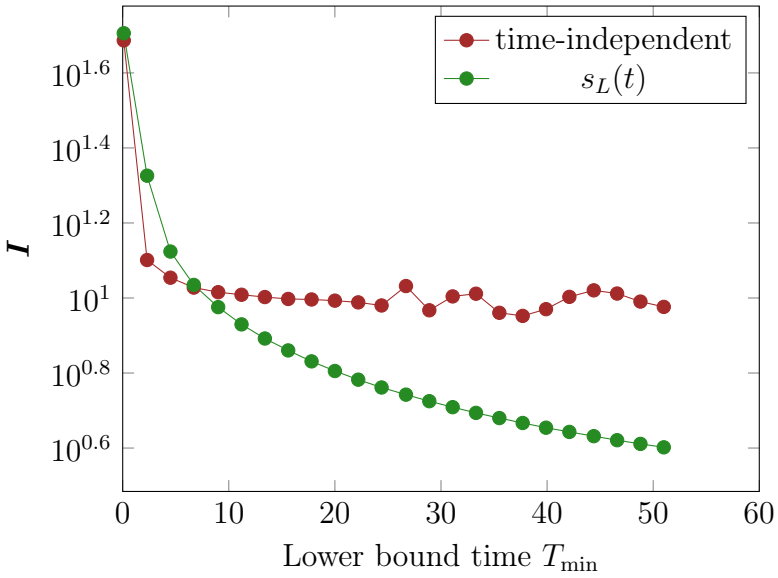


Figure 2.9: ***I* for increasing lower bound time $T_{\min}$.** The plot shows I for increasing values of lower bound time $T_{\min}$, using the time-independent Hamiltonian (red) and time-dependent Hamiltonian (green) with linear $s_L(t)$ and evaluated for a Cy(51). This distribution reflects the probability distribution of the two approaches: for the time-independent Hamiltonian the probability does not increase with time, resulting in a (almost) constant I, while the time-dependent Hamiltonian showing localization properties requires less iterations to get to unitary probability. Note that the plot is given with logscale y axis, which helps to highlight the time-independent trend.

The iterations trend reflects the overall probability distribution of the time-dependent and time-independent Hamiltonian approaches. For small lower bound time the two approaches show a similar performance: the probability is very small, thus requiring a large number of iterations to get to unitary probability. As $T_{\min}$ increases we see two very different trends:

- The time independent approach requires an almost constant number of iterations[6]. This reflects the non-localization properties of this particular approach, for which the probability does not increase with time. It also shows that the maximum probability found is (almost) equal for all T, provided that $T_{\min}$ is large enough.

[6]The numerical value of I is irrelevant since it reflects only the Cy(51).

- On the other hand the time-dependent approach requires less iterations to solve the search with unitary probability, as expected.

Taking into account the multiple runs and the initialization time t_{init} previously discussed, it is clear that the time-depedent approch performs better than the time-independent counterpart in most of the scenarios, where imposing a lower bound time $T_{\min}$ becomes necessary.

τ and run iterations with constrained lower bound time

In order to show that the lower bound time does indeed have such a great impact on the performance of the time-dependent approach relative to the time-independent one, we study the distribution of τ and I with a constraint on $T_{\min}$. It should be noted that in this particular scenario τ will no longer be a minimization over T since the time is fixed, while the dependance on γ remains.

The choice of constraint is arbitrary and somewhat biased since the larger the constraint the better the performance of the time-dependent approach, as we have just shown in Figure 2.8. Therefore, to make the choice fair, we consider the lower bound time to be $T_{\min} = \frac{\pi}{2}\sqrt{N}$ which is the same order of magnitude of the standard Grover's time scaling, the time-independent quantum walk search on the complete graph and the unstructured local adiabatic search. In the best-case scenario we might discover that the number of iterations necessary to get to unitary probability remains constant regardless of the dimension of the graph, making this approach have the same time scaling as the ones just mentioned; in the most probable scenario we discover that the number of iterations increases with the graph size, thus adding a scaling factor that depends on some power of N.

We begin by investigating the effects of the interpolating schedule $s(t)$ on the τ distribution. Figure 2.10 shows the results for cycle graphs $Cy(N)$ with N up to 71, using the time-dependent Hamiltonian and the interpolating schedules defined in Section 2.1.2.

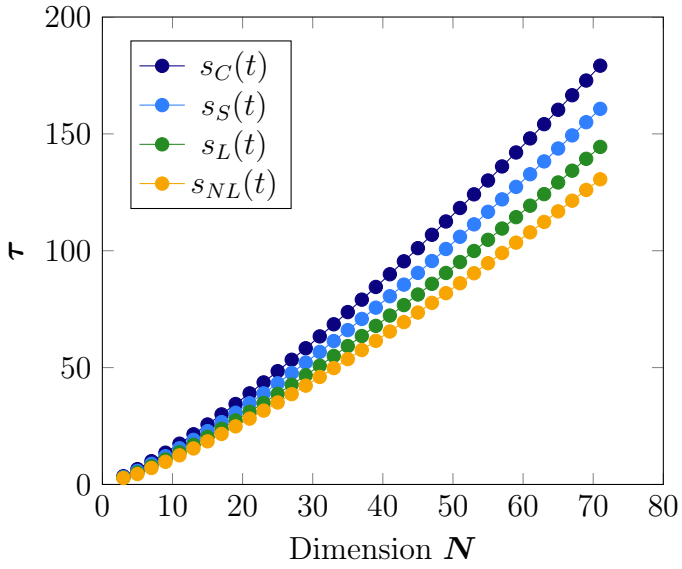


Figure 2.10: **τ with constrained time at $T = \pi/2\sqrt{N}$, for the different intepolating schedules.** The plot shows τ for cycle graphs up to $N = 71$, using the time-dependent Hamiltonian with different intepolating schedules: (orange) non-linear $s_{NL}(t)$, (green) linear $s_L(t)$, (light blue) $s_S(t)$ and (blue) $s_C(t)$. As expected $s_{NL}(t)$ is the best performing intepolating schedule, followed by the linear $s_L(t)$. The others do not show any advantage.

As we could have expected the non-linear interpolating schedule $s_{NL}(t)$ performs the best, compared to the linear one. The $s_S(t)$ and $s_C(t)$ show poor performance compared to all the others, therefore are ignored for the next analysis. This shows indeed that the shape of the interpolating function has a great impact on the performance of the algorithm, in particular when considering large graphs.

We now compare the time-independent approach to the time-dependent one using the linear and non-linear interpolating schedules. Again, the time is constrained as in the previous case. We begin by looking at the τ.
In Figure 2.11 we can divide the plot into two sections. For small graph dimension $N < 20 - 25$ the time-dependent approach performs comparably to the time-independent one, although the latter has a slight advantage. On the other hand, for large graphs the time-dependent search distribution deviates considerably from the time-independent, for both the linear $s_L(t)$ and Roland-Cerf $s_{NL}(t)$ interpolating schedules. This is indeed a consequence of the lower bound time $T_{\min}$ that we considered, begin dependent on the dimension of the graph and increasing with N. Nevertheless, for large N the mamimum probability found for the

time-independent approach decreses, while for the time-dependent Hamiltonian it increases making it a succesfull alternative for larger graph dimensions.

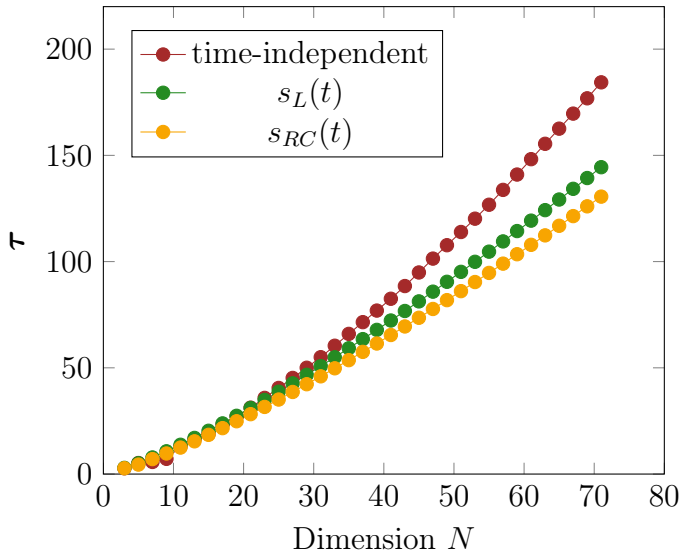


Figure 2.11: **τ with constrained time at $\pi/2\sqrt{N}$, for the time-dependent and time-independent approaches.** The plot shows τ for cycle graphs up to N=71, using the time-independent Hamiltonian (red) and time-dependent Hamiltonian with linear s_L (green) and non-linear s_{NL} (orange) interpolating schedules. We notice that for small graphs, up to $N \approx 20$, both approaches perform similarly. For large N however the time-dependent algorithm performs significanly better.

We now turn our attention to the run iterations. Figure 2.12 shows the number of iterations I for the two considered approaches. We are able to qualitatively describe the time scaling of the algorithms by superimposing the function $\frac{2}{\pi}\sqrt{N}$ (black line): if the plot of the iterations I lies below the line the time-dependent algorithm performs better than the classical search. Remember in fact that we imposed a lower bound time $T_{\min} = \pi/2\sqrt{N}$, and combining it with $2/\pi\sqrt{N}$ gives us the time scaling of the classical search algorithm $O(N)$.

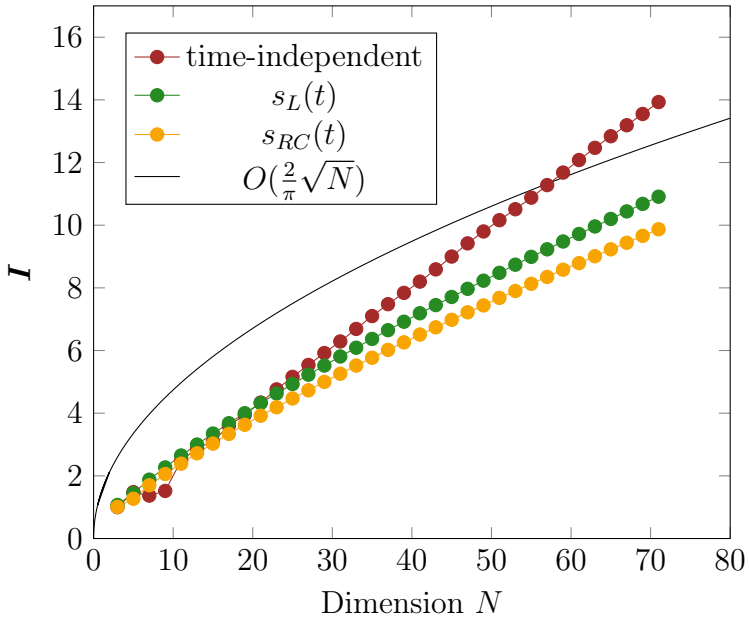


Figure 2.12: ***I* with constrained time at $\pi/2\sqrt{N}$, for the time-dependent and time-independent approaches.** The plot shows I for cycle graphs up to $N = 71$, using the time-independent Hamiltonian (red) and time-dependent Hamiltonian with linear s_L (green) and non-linear s_{NL} (orange) interpolating schedules. The black line represents a time scaling of $\sqrt{N}$. Combining the value of I with the constrained time of $\pi/2\sqrt{(N)}$ we can estimate the performance of the algorithms: if the distribution is below the black line the algorithm performs better than the classical search $O(N)$. From this plot we can clearly see that for N up to 71 the performance is better than the classical search, although we are not able to make predictions for larger N.

Clearly this plot reflects what we have just seen for τ distribution, but it is able to give an immediate idea on the number of run iterations necessary to solve the search problem. For all the graph considered - with N up to 59 - both the time-dependent and time-independent perform better than the classical search algorithm. However, for $N > 59$ and up to 71, the time-independent approach starts to perform worse than the classical counterpart, while the time-dependent still has a better performance. It it difficult however to predict with confidence the trend for larger graph. Nevertheless the time-dependent approach, in particular with the Roland-Cerf $s_{NL}(t)$ interpolating schedule shows promising results.

2.4.5 Comparison: robustness

We now address the robustness of the time-dependent and time-independent search. As we mentioned in Section 2.3.2 we are only interested in the comparison of the two approaches, and not an absolute measure of their robustness. Therefore we will use this measure as a comparison value.
We proceed by evaluating the γ-robustness considering small variations on the parameter γ. Recalling that we evaluated the probability on a grid, we consider the variations in the order of 2 *square*. We begin by finding the quantity τ with time constrains $T = \pi/2\sqrt{N}$. For the corresponding (T, γ) combination we evaluate the robustness R_γ. For the time-dependent search with only consider the linear interpolating schedule and the Roland-Cerf $s_{NL}(t)$. As we showed in the previous section the Roland-Cerf Hamiltonian performs better than the linear conterpart, while from a qualitative point of view the linear Hamiltonian has a smoother probability distribution (see Figure 2.4(a)-(d)). The following plots show the γ-robustness for the time-independent and time-dependent search.

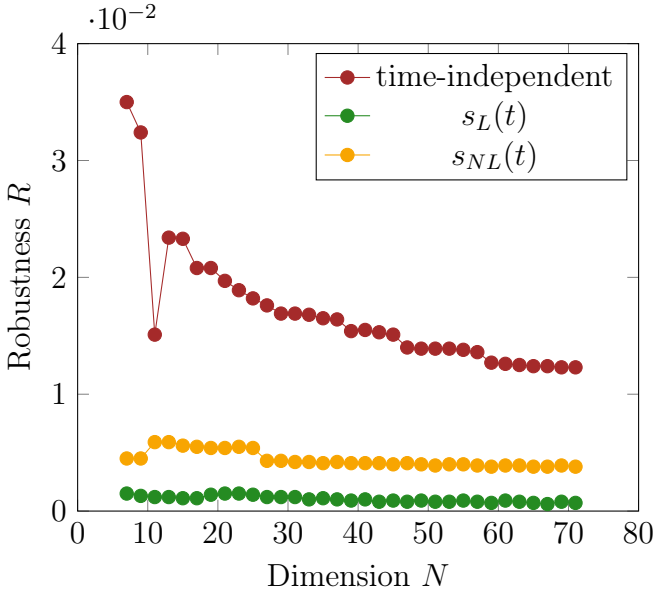


Figure 2.13: **γ-Robustness for the time-independent and time-dependent approaches.** The figure shows the γ-robustness for the time-independent approach (red), the time-dependent one with linear $s_L(t)$ (green) and non-linear $s_{NL}(t)$ (red) interpolating schedules. Recalling that the lower R value the highest the robustness, this distribution reflects the probability seen in Section 2.4.2, where the probability distribution was smoother for $s_L(t)$ than $s_{NL}(t)$.

Indeed Figure 2.13 shows that the time-dependent approach is more robust than the time-independent one. In particular, the linear interpolating schedule

$s_L(t)$ leads to a greater robustness compared to the non-linear $s_{NL}(t)$. It is to be noted however that the difference in robustness between the time-independent and time-dependent approach is much greater than the difference between the interpolating schedules.

We now turn our attention the T-robustness, evaluated similarly to the γ-robustness. From Figure 2.14 we see that, in contrast with the γ counterpart, the time-indipendent approach performs better compared to the time-dependent one. However, if we compare the R values with the R_γ values we notice that the difference between the two approaches is much smaller for the T-robustness. For this reason we can safely say that the search with time-dependent Hamiltonian is more robust than the time-independent one. Additionally, the improved performance acheived with the non-linear interpolating schedule $s_{NL}(t)$ comes at the cost of less robustness.

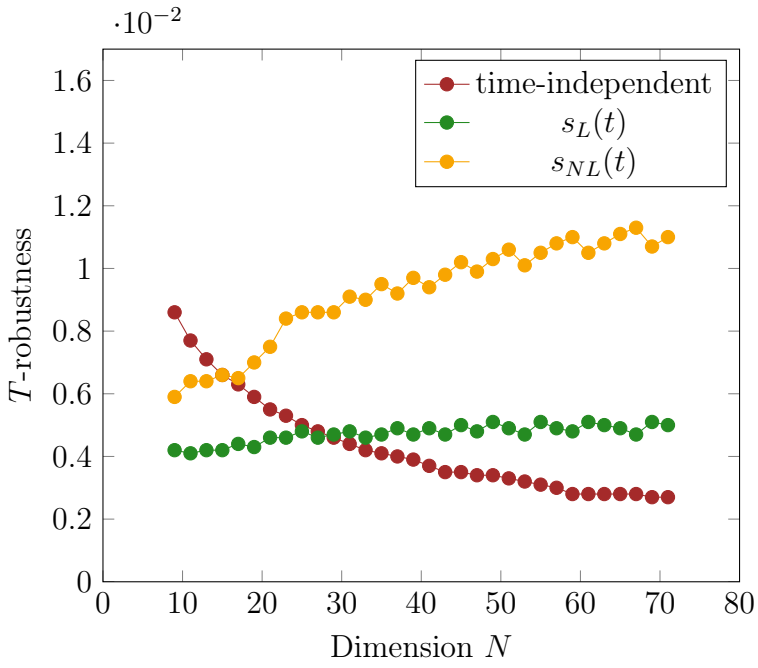


Figure 2.14: ***T*-Robustness for the time-independent and time-dependent approaches.** The figure shows the T-robustness for the time-independent approach (red), the time-dependent one with linear $s_L(t)$ (green) and non-linear $s_{NL}(t)$ (red) interpolating schedules. Surprisingly, the time-independent approach is more robust than the time-dependent one for large N. However it is to be noted that the difference in values is much smaller than the one obtained for the γ-robustness.

2.5. Results for the complete graph

We now turn our attention to the complete graph. As we discussed in the preliminaries in Section 1.4.1 and Section 1.5.3, with the complete graph we are able to solve the search problem using the standard time-independendent quantum walks Hamiltonian with time scaling of $O(\sqrt{N})$. Additionally for the unstructured search - which is equivalent to a search on the complete graph - we showed that with the local adiabatic evolution it is possible to get the same speedup of $O(\sqrt{N})$, while that was not the case for the global adiabatic evolution that had the same time scaling as the classical search. Although Wong proved that for the complete graph it is not possible to solve the search problem with an adiabatic quantum walk algorithm, for completeness we extend the time-dependent Hamiltonian implementation to the complete graph. Clearly we are not able to achieve any speedup nor necessarily any comparable time scaling, but we might get some interesting insights in terms of probability distribution and robustness.

2.5.1 Comments on the placement of γ

Firstly we recall that in Section 1.1.2 we introduced the search Hamiltonian as

$$H = \gamma L - |w\rangle\langle w| \tag{2.12}$$

We quickly notice that the γ parameter is in front of the Laplacian, compared to the time-dependent Hamiltonian considered throughout our work where γ was placed in front of the oracle $|w\rangle\langle w|$. We could now proceed by comparing the time-independent and time-dependent approaches with the γ placement as in literature - i.e. in front of the Laplacian - or as we did in **??**. In order to be consistent with the standard quantum walks search on the complete graph discussed in Section 1.4.1 we proceed by considering the following Hamiltonians, with γ in front of the Laplacian. This ensures that the time-dependent probability distribution is compatible with the time-independent one, and does not require to re-evaluate the optimal γ for the time-independent approach. Nevertheless the two approaches can be considered equivalent. The Hamiltonians are given by

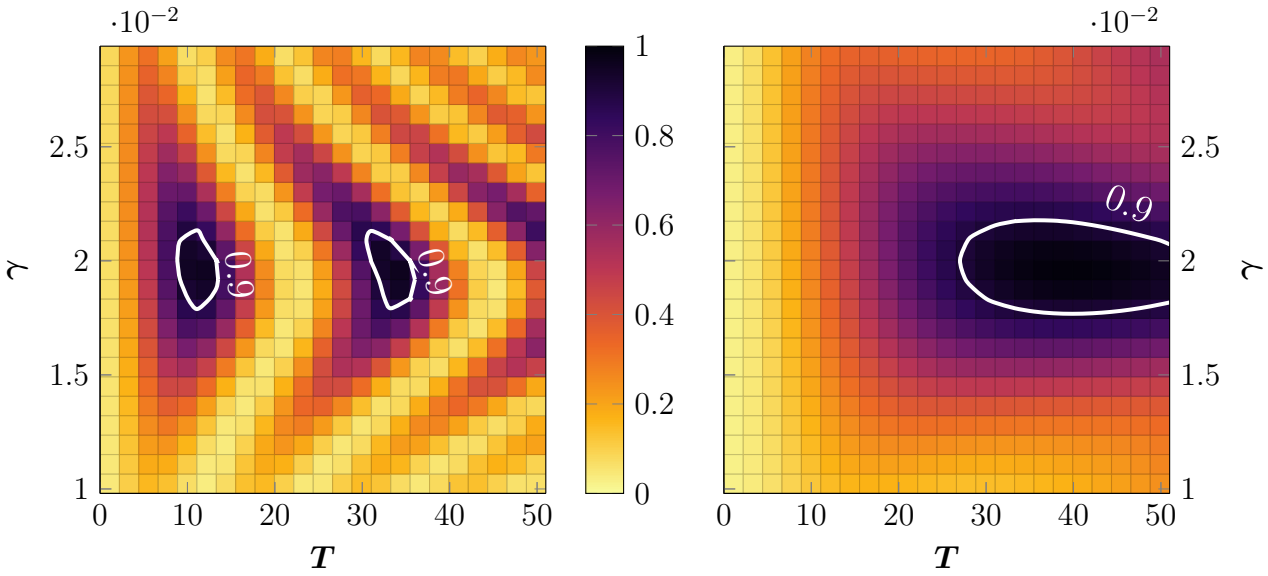
$$H = \gamma L - |w\rangle\langle w| \qquad \text{(independent)} \tag{2.13}$$

$$H(s) = (1-s)\gamma L - s|w\rangle\langle w| \qquad \text{(dependent)} \tag{2.14}$$

Having defined the Hamiltonians, we proceed as in the previous section, following however a more qualitative approach. In order to do so we compare the probability distribution using the heatmap plots introduced in **????**.

2.5.2 Probability distribution and qualitative robustness

Recalling that for the time-independent quantum walks implementation the optimal γ is given by $\gamma = 1/N$, we evaluate the probability for γ in a neighbourhood of $1/N$ and up to $T = N$. This allows us to have a complete picture of the probability distribution, as can be seen in the following heatmap plots.



(a) time-independent (b) $s_{NL}(t)$

Figure 2.15: **Probability distributions for the complete graph $C(51)$ with time-dependent and time-independent Hamiltonians.** The two figures show the probability distribution for the complete graph $C(51)$ with the time-independent Hamiltonian (a) and time-dependent Hamiltonian (b) using the non-linear $s_{NL}(t)$ interpolating schedule. The white contour lines indicate a region with probability $p > 0.9$.

As previously discussed we are able to solve the search problem with the time-independent algorithm in a time of the order of $T = \pi/2\sqrt{N}$. Indeed we see from the figure above that the probability is close to $p = 1$ for $T = 11$, as expected. On the other hand the time-dependent approach is able to solve the search with a single iteration for $T \to N$, showing no significant speedup compared to the

classical search. This however leaves space for a multiple run search as introduced in Section 2.1.3. We therefore investigate this possibility by evaluating the maximum probability found for $T = \pi/2\sqrt{N}$ with the time-dependent algorithm and study the **iterations** distribution.

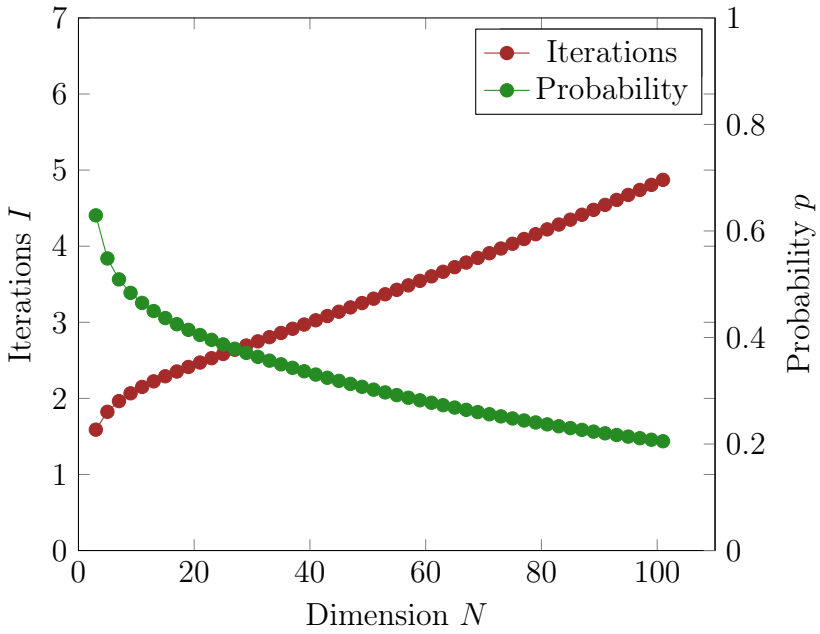


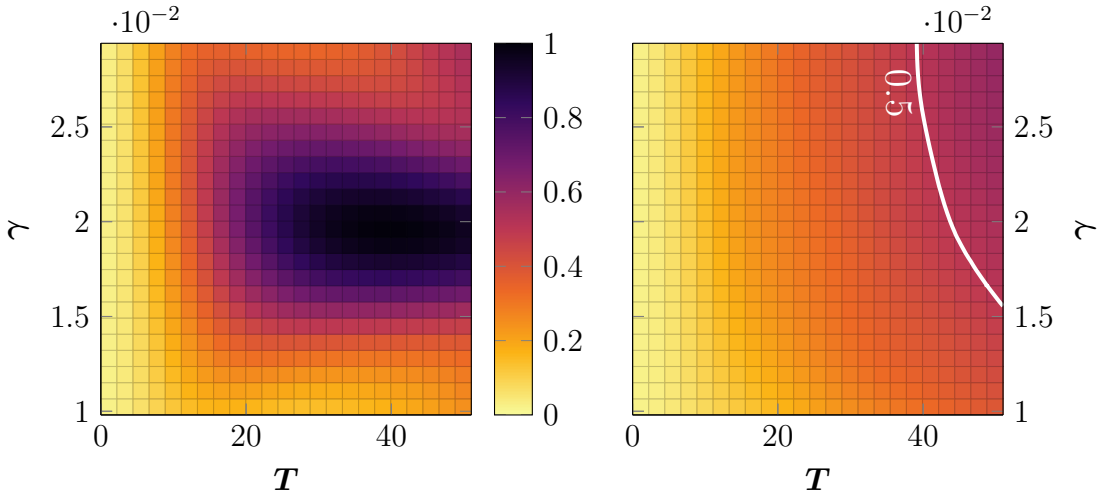
Figure 2.16: **Iterations I and probability p for the multiple run search on complete graph.** The plot shows the probability p and the number of iterations I for the multiple run search with the time-dependent Hamiltonian. The time is constrained to $\pi/2\sqrt{N}$ as in the solution of the standard quantum walks search. It is clear that the number of iterations increases linearly, although very slowly. Notice in fact that N goes all the way up to $N = 101$. For the limit of large N the time-dependent approach is therefore not a valid alternative, nor of comparable performance.

It is clear from the plot that the number of run iterations increases with the graph size linearly, although very slowly (notice that N goes up to 101). In the limit for large N the multiple runs search does not improve the time-dependent approach, therefore the standard quantum walks search remains the strongest option.

However, we can now address the **robustness** properties of the two approaches. Similarly to the cycle graph, the complete graph has a probability distribution with regions of high and low probability when considering the time-independent Hamiltonian. Compared to the smooth probability distribution of the time-dependent algorithm we can safely say that the latter is more robust than the first, both for noise on T and γ. It is to be noted however that for the cycle graph the time-independent and time-dependent approaches are of comparable

performance, therefore the robustness has a great impact to the overall performance. In this scenario the time-independent approach performs better in terms of probability, and having less robustness does not justify the *almost* linear time-scaling of the time-dependent one.

Although the time-dependent algorithm is not able to perform as well as the time-independent one, striking is the difference in probability distribution between the non-linear $s_{NL}(t)$ and the linear $s_L(t)$, showing the importance of the shape of the interpolating schedule. From the following plots we can see that the time-dependent Hamiltonian with the linear interpolating schedule $s_L(t)$ is not able to solve the search for $T = N$. The improvement on the performance is therefore achieved by findind the optimal interpolating schedule.



(a) $s_{NL}(t)$ (b) $s_L(t)$

Figure 2.17: **Probability distributions for the complete graph $C(51)$ with the time-dependent Hamiltonian.** The figure shows the probability distribution for a complete graph of $N = 51$ using the time-dependent Hamiltonian with the non-linear $s_{NL}(t)$ (a) and linear $s_L(t)$ (right) interpolating schedules. It illustrates the great impact of the interpolating schedule on the overall performance of the time-dependent algorithm, where the choice of $s(t)$ is critical.

Conclusions

We conclude our work with a summary of what has been accomplished in this thesis, some thoughts on the results and future perspectives.

In Chapter 1 we reviewed some basic notion of graph theory, quantum walks, and the main characteristics of the graph topologies considered. We then introduced the search problem as originally posed by Grover - with particular emphasis on the action of the *oracle* - followed by its quantum walks implementation of Farhi and Gutmann on the complete graph. We then discussed the adiabatic theorem and its application to the search problem, focusing on the difference between the global and local adiabatic evolution. To set the basis for our work we presented the study by Wong *et. al* [13], where they showed that an adiabatic-quantum walk implementation of the search algorithm is not possible with the structure of the standard Grover's oracle.

In Chapter 2 we introduced the main topic of our work which is a quantum walks search algorithm with time-dependent Hamiltonians, *inspired* by the adiabatic implementation but free of the constrains of the adiabatic theorem. We introduced a few classes of interpolating schedules with the goal of improving the standard linear one of Farhi and Gutmann. We draw from the one derived by Roland and Cerf, and consider the following non-linear interpolating schedule:

$$s_{NL}(t) = \frac{1}{2}\Big[\big(2\frac{t}{T} - 1\big)^3 + 1\Big].$$

Additionally we also consider the possibility of repeating the search multiple times if the search is not perfect - that made us take into account an initialization and measure time. In order to compare the time-independent and time-dependent approach we introduced three classes of results: the search, the localization and the robustness.

We then turned our attention to the cycle graph, the main graph topology considered. We studied the localization, search and robustness.

In terms of localization we discovered that the probability evaluated with the time-independent Hamiltonian does not increase with time, therefore it does not show localization properties. On the other hand, the time-dependent approach - given that it is based on the adiabatic implementation - for large T (far larger than the classical $O(N)$ search) it is able to achieve unitary probability. More interestingly, since the probability does not increase linearly, the solution of the search can be found with probability in the order of $p = 0.8 \div 0.9$ in much less time.

We then studied the search performance of the two approaches in terms of multiple runs search. Therefore we introduced a new quantity which represents the minimum time necessary to get to unitary probability:

$$\tau = \min \left(\frac{T}{p} \right)_{T,\gamma}.$$

We also consider the number of run iterations $I = \min(p^{-1})_{T,\gamma}$. Additionally we discover that τ requires to consider a minimum time $T_{\min}$ to be effective at comparing the two approaches. To be consistent with the standard Grover's and the quantum walks search we set $T_{\min} = \pi/2\sqrt{N}$.

After seeing that for the time-dependent approach the linear and non-linear interpolating schedules s_L and s_{NL} perform the best, we compare them to the time-independent approach.
We discovered that both approaches have similar performance up to $N \approx 25$, while for large N it gets significanly different, with the s_{NL} performing more efficiently than the time-independent approach. In terms of run interations I we see that the time-dependent approach, regardless of the interpolating schedules, performs better than the classical search, but still much worse than the optimal time scaling of $O(\sqrt{N})$. Although promising, this result is limited to the dimension considered - up to $N = 71$ in our analysis - and therefore cannot be generalized for large N.

We then studied the robustness for both time and γ. The time-independent approach has a very discontinous probability, made of scattered regions of high

and low probability, leading to being less γ-robust than the time-dependent approach. Indeed the latter has a smooth probability distribution regardless of the interpolating schedule considered. Nevertheless the linear s_L leads to more robust results than s_{NL}. In terms of T-robustness the time-independent approach is surprisingly more robust than the others, although the difference in robustness is much smaller than the one encountered for the γ-robustness. Therefore we can safely say that the time-dependent Hamiltonian-based algorithm is more robust than the time-independent one.

For completeness we at last turned our attention to the complete graph, qualitatively comparing the probability distribution and the robustness. As expected the time-dependent approach is not able to achieve comparable performance with the standard time-independent algorithm. In terms of qualitative robustness we see that the time-dependent algorithm has a smoother probability distribution and therefore better robustness. However the improvement in robustness does not justify the much worse time scaling.

The complete graph however illustrates the importance of the interpolating schedule. In particular, with the time-dependent approach and linear s_L the maximum probability reached is $p = 0.5$ in $T = N$ for a $C(51)$. With the improved non-linear interpolating schedule s_{NL} we are able to achieve in $T = N$ a maximum probability of $p > 0.9$.

Although it is not able to achieve the same time scaling as the time-independent approach it suggests that the improvement on performance comes from the choice of the optimal interpolating schedule. Future investigation on the interpolating schedule can lead to improvements on performance, in particular for the cycle graph. Additionally one could also apply the time-dependent Hamiltonian to other graph topologies.

Bibliography

[1] G. Brassard, P. Høyer, and A. Tapp. Quantum Algorithm for the Collision Problem. *Encyclopedia of Algorithms*, pages 1–2, 2015.

[2] A. M. Childs, E. Farhi, and S. Gutmann. An example of the difference between quantum and classical random walks. *Quantum Information Processing*, (2):1–4, 2001.

[3] A. M. Childs and J. Goldstone. Spatial search by quantum walk. *Physical Review A - Atomic, Molecular, and Optical Physics*, 70(2), 2004.

[4] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. Quantum Computation by Adiabatic Evolution. *arXiv: Quantum Physics*, 2000.

[5] E. Farhi and S. Gutmann. Analog analogue of a digital quantum computation. *Physical Review A - Atomic, Molecular, and Optical Physics*, 57(4):2403–2406, 1998.

[6] M. Fürer. Solving np-complete problems with quantum search. In E. S. Laber, C. Bornstein, L. T. Nogueira, and L. Faria, editors, *LATIN 2008: Theoretical Informatics*, pages 784–792, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[7] L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letters*, 79(2):325–328, 1997.

[8] J. G. Morley, N. Chancellor, S. Bose, and V. Kendon. Quantum search with hybrid adiabatic?quantum walk algorithms and realistic noise. *Physical Review A*, 99:1–24, 2018.

[9] O. Mülken and A. Blumen. Continuous-time quantum walks: Models for coherent transport on complex networks. *Physics Reports*, 502(2-3):37–87, 2011.

[10] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambrigde University Press, 2000.

[11] J. Roland and N. J. Cerf. Quantum search by local adiabatic evolution. *Physical Review A - Atomic, Molecular, and Optical Physics*, 65(4):6, 2002.

[12] S. Z. SH. Hung, S. Hietala. Quantitative Robustness Analysis of Quantum Programs (Extended Version). *Proceedings of the ACM on Programming Languages*, 3(January), 2019.

[13] T. G. Wong and D. A. Meyer. Irreconcilable difference between quantum walks and adiabatic quantum computing. *Physical Review A*, 93(6):1–8, 2016.