

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 spatial search problem on graph**

**graphs  
a graph**

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

## Abstract

In this thesis we study the properties of quantum walks with time dependent hamiltonian, focusing in particular on the application to the quantum search problem on **graphs** graph. We compare the time independent and the time dependent approach for two graph topologies, the circular and the complete graph, which represent respectively the worst and best case scenario for the known quantum search. We also investigate the role of the function that regulates the time-dependance of the hamiltonian in the time scaling at which the solution is found. Lastly we exploit the consequences of the adiabatic theorem to study the localization properties of the time-dependent approach.

# Contents

<b>Introduction</b>	<b>6</b>
<b>1 Preliminaries</b>	<b>7</b>
1.1. Graph Theory . . . . .	7
1.2. Quantum Walks . . . . .	8
1.3. Grover's Quantum Search . . . . .	8
1.4. Quantum Search with Continuous-Time Quantum Walk . . . . .	9
1.4.1 Search on Complete Graph . . . . .	10
1.5. Adiabatic Theorem . . . . .	10
1.5.1 Computation by Adiabatic Evolution . . . . .	11
1.6. Impossibility of Adiabatic Quantum Walks . . . . .	11
<b>2 Quantum Walks with Time-Dependent Hamiltonians</b>	<b>12</b>
2.1. Search with Time-Dependent Hamiltonian . . . . .	12
2.1.1 Time-Dependent Quantum Walk . . . . .	13
2.1.2 Interpolating Schedule $s(t)$ . . . . .	13
2.1.3 Multiple Runs for One Search . . . . .	14
2.2. Selected Topology: Circular and Complete Graph . . . . .	15
2.3. Characterization of the results: Search, Localization and Robustness . .	16
2.3.1 Search vs Localization . . . . .	16
2.3.2 Robustness . . . . .	16
2.4. Results for the Circular Graph . . . . .	17
2.4.1 Time-Independent Benchmarks . . . . .	17
2.4.2 Time-Dependent Results . . . . .	18

---

2.4.3	Comparison: Localization . . . . .	20
2.4.4	Comparison: Search . . . . .	20
2.4.5	Comparison: Robustness . . . . .	25
2.5.	Results for the Complete Graph . . . . .	26
2.5.1	Search results from Wong(2016) . . . . .	26
2.5.2	Localization results . . . . .	26
	<b>Conclusions</b>	<b>27</b>
	<b>Appendices</b>	<b>28</b>
	Appendix A: Probability Grid Evaluation . . . . .	28
	2.5.3 Optimization Algorithm . . . . .	28
	2.5.4 Schroedinger Solver and Normalization Error . . . . .	29
	Appendix B: Computational Routines . . . . .	29
	<b>Bibliography</b>	<b>30</b>

# List of Figures

1.1	Pictorial representation of circular and complete graph . . . . .	8
2.1	Ronald and Cerf's interpolating schedules for the unstructured search and our non-linear schedule . . . . .	14
2.2	Probability heatmap plot for the time-independent hamiltonian, N=47 .	18
2.3	Probability heatmap plot for the time-dependent hamiltonian, for different shapes of $s(t)$ . . . . .	19
2.4	$t_f/p$ distribution for sampled $\gamma$ showing that the minimum will always be for the smallest $t$ . . . . .	21
2.5	$\min(t_f/P)$ distribution for increasing lower bound time. . . . .	22
2.6	$iters$ distribution for increasing lower bound time. . . . .	23
2.7	$\min(t_f/P)$ distribution for Cy(N) up to N=71 with constrained time at $\sqrt(N)$ . . . . .	24
2.8	$\min(t_f/P)$ distribution for Cy(N) up to N=71 with constrained time at $\sqrt(N)$ and different interpolating schedules. . . . .	25

# Chapter 1

## Preliminaries

### 1.1. 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, which represent the connection between any two pair of vertices. If indeed any two vertices  $(i, j)$  are connected by an edge we define as **adjacent**, and from this we can construct the *adjacency matrix*  $A$  as:

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

which represents the connectivity of the graph. We can then describe the degree of each vertex of the graph, namely the number of vertices (excluding itself) connected to it, through the *degree diagonal matrix*  $D_{jj} = \deg(j)$ .

It is also useful to introduce the *laplacian matrix* defined as **matuscolo**

$$L = A - D \quad (1.2)$$

which ~~we'll later see~~ describes the quantum walk evolution.

In the quantum realm a vertex ~~is identified with a~~ a vector in an  $N$  dimensional Hilber space, so that a vertex  $|j\rangle$  can be represented in the following way

$$|j\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \text{questo è } |1\rangle \text{ non } j \quad (1.3)$$

Throughout this work we focus our attention on two particular graph topologies: the **circular** graph and the **complete** graph (FIG. 1).

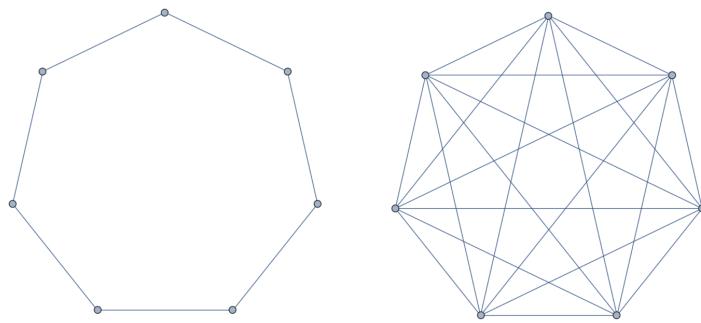


Figure 1.1: Pictorial representation of a circular graph (left) and a complete graph (right) with  $N=7$

## 1.2. Quantum Walks

The continuous-time quantum walk is the direct analogue of the classical random walk. Given a graph  $G$ , a random walk is a Markov process with a fixed probability for unit time  $\gamma$  of jumping to an adjacent vertex  $j$ . This particular walk can be described by a linear differential equation in terms of probability, namely

$$\frac{d}{dt} p_j(t) = \gamma \sum_k L_{jk} p_k(t) \quad (1.4)$$

where  $p_j(t)$  is the probability of being on the vertex  $j$  at time  $t$ .

**controllare se analogue si dice**  
The quantum analogue takes place in an  $N$ -dimensional Hilbert space spanned by the states (vertex)  $|j\rangle$  of the graph  $G$ . **Instead of considering the classical probability as we previously did**, we consider the probability time-dependent amplitudes  $q_j(t) = \langle j|\psi(t)\rangle$ , where  $|\psi\rangle$  is a general time-dependent state. The differential equation takes thus the **viene dall'eq. di Schrödinger** form of

**manca hbar.**  
**Possiamo porre hbar=1**

$$i \frac{d}{dt} q_j(t) = \sum_k H_{jk} q_k(t) \quad (1.5)$$

The continuous-time quantum walk is defined by letting  $H = -\gamma L$ , where  $L$  is the **previously defined** Laplacian matrix. **defined in Eq (...).**

## 1.3. Grover's Quantum Search**serve?**

This section should include the standard Grover's Quantum Search to give the context for the quantum walk approach.

## 1.4. Quantum Search with Continuous-Time Quantum Walk

[referenza al paper pubblicato](#)

*Spatial Search by Quantum Walk, A. Childs, J. Goldstone, quant-ph/0306054v2*

io non  
parlerei fi  
grover.  
introdurrei  
direttamente  
l'algoritmo  
di childs

We now address the quantum search problem firstly formulated as Grover's algorithm and then extending it to the search on a graph using quantum walks.

To approach the Grover problem with quantum walk it's necessary to modify the hamiltonian such that the vertex  $|w\rangle$ , i.e. the target, is somewhat special. Following Grover's oracle an oracle hamiltonian  $H_w$  is introduced

$$H_w = -|w\rangle\langle w| \quad (1.6)$$

which in particular has energy zero for all but the vertex  $|w\rangle$  for which it has enenergy in some units..

-1. Therefore the Grover problem, i.e. quantum search, becomes finding the ground state of such hamiltonian. To do so we consider the time-independent hamiltonian of the form

$$H = -\gamma L + H_w = -\gamma L - |w\rangle\langle w| \quad (1.7)$$

where  $L$  is the laplacian of the graph, which contains the information of the dynamics over that particular graph topology. The evolution of the quantum walk is therefore governed by this hamiltonian.

### algorithm

The quantum search routine works as follow:

- we consider the superposition of all possible states, namely

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_j |j\rangle \quad (1.8)$$

### Hamiltonian $H$

- we find the evolved state using the hamiltonian for a time  $T$   $H$

$$|\psi(T)\rangle = U(T)|s\rangle = \exp\left\{-\frac{i}{\hbar}HT\right\}|s\rangle \quad (1.9)$$

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

- we then measure the occupation probability of the target

$$p = |\langle w|\psi(T)\rangle|^2 \quad (1.10)$$

The objective is to find the optimal value of  $\gamma$  so that the probability of the system of finding itself in  $|w\rangle$  is as close as possible to 1 for the smallest  $T$  possible

### 1.4.1 Search on Complete Graph

We now look at the search on a complete graph. This case is particularly interesting since it can be solved analitically

to be continued

## 1.5. Adiabatic Theorem

*Quantum Computation by Adiabatic Evolution, E. Farhi, J. Goldstone, S. Gutmann, M. Sipser, quant-ph/0001106 paper pubblicato*

A quantum system evolves according to the Schroedinger equation

$$i \frac{d}{dt} |\psi(t)\rangle = H(t) |\psi(t)\rangle \quad (1.11)$$

and defining the instantaneous eigenstates and eigenvalues of  $H(t)$  by

$$H(t)|l; t\rangle = E_l(t)|l; t\rangle \quad (1.12)$$

invertire i segni

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

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

$$\lim_{T \rightarrow \infty} |\langle l=0; t=T | \psi(T) \rangle| = 1 \quad (1.13)$$

This means that if the system is chosen to evolve at a slow enough rate, the instantaneous hamiltonian will remain in the ground state through the evolution. It is useful to consider a smooth one-parameter hamiltonian  $H(s)$  such that  $s = t/T$ , with  $t \in [0, T]$  so that  $s \in [0, 1]$ . Let's now define the energy minimum gap by

$$g_{min} = \min_{0 \leq s \leq 1} (E_1(s) - E_0(s)) \quad (1.14)$$

In addition we can find a time lower bound  $T^*$  such that for  $T \gg T^*$  the probability is arbitrarily close to 1, in detail

$$T \gg \frac{\varepsilon}{g_{min}^2} \quad (1.15)$$

where

$$\varepsilon = \max_{0 \leq s \leq 1} \left| \left\langle l=1; s \left| \frac{dH(s)}{dt} \right| l=0; s \right\rangle \right| \quad (1.16)$$

d/ds

### 1.5.1 Computation by Adiabatic Evolution

Let's now discuss how to take advantage of the adiabatic theorem introducing the usual way in which the adiabatic evolution is implemented. It is often presented 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 for easily preparing the state to be evolved. The adiabatic implementation then consists, assuming that the ground state of  $H_P$  is unique, in having a time dependent hamiltonian  $H(s)$  such that

$$H(s) = (1 - s)H_B + sH_P \quad (1.17)$$

In this way we can prepare for  $s = 0$  the system in  $H_B$  and let it evolve so that for  $s = 1$  it reaches  $H_P$ . Thanks to the adiabatic theorem, if it's made to evolve sufficiently slowly we will find ourselves in the ground state of the problem hamiltonian, which is exactly the solution.

## 1.6. Impossibility of Adiabatic Quantum Walks

[6]

Chapter **2**

# Quantum Walks with Time-Dependent Hamiltonians

The main purpose 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 topology: the circular 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 optimized-search, localization - which represents a search without needs to optimize the time - and a measure of robustness.

## 2.1. Search with Time-Dependent Hamiltonian

In Section? we discussed the use of quantum walks for the spatial search problem [1] 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 [2]. Efforts to combine the two approaches showed that the search problem can be solved only with structures stronger than the usual Grover's oracle [6], 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 Walk

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  $H_w = \gamma |w\rangle\langle w|$

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

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

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:

$$S(t, t_0) = T \exp \frac{1}{i\hbar} \left\{ \int_{t_0}^t dt' H(t') \right\} \quad (2.2)$$

metti uno spazio

attenzione che  
usi una  
notazione  
mista con e  
senza hbar

where  $T$  is the time-ordering. Since we're only interested in the evolved state, having the exact evolution operator is somewhat irrelevant. Additionally for the circular graph - which represents the main topology studied in this thesis - the search hamiltonian does not show any particular properties making an analitical derivation of the operator not possible. We therefore proceed by solving the differential Schrödinger equation:

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

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

$$\frac{d}{dt} |\psi_i(t)\rangle = \sum_j H_{ij} |\psi_i(t)\rangle \quad \text{\textcolor{red}{\backslash psi\_j?}}$$

with the boundary conditions  $|\psi_i(0)\rangle = |\psi_i\rangle$ . chi sono gli psi\_i?

From a numerical point of view this approach is also much simpler than other direct methods of evaluating the evolution operator, such as the Dyson Series or Magnus Expansion. are you sure?

### 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. [2] uses a linear interpolating schedule defined as  $s(t) = \frac{t}{t_f}$ . Ronald and Cerf show that in order to obtain a quadratic speedup for the complete graph a non linear schedule is essential [4][3].

scrivertlo  
meglio

consider also

Thus, from a linear interpolating schedule we begin considering, somewhat naively, quadratic and cubic schedules:

$$s(t) = \sqrt{\frac{t}{t_f}} \quad s(t) = \sqrt[3]{\frac{t}{t_f}} \quad (2.5)$$

We then consider the interpolating schedule analitically derived by Ronald and Cerf for the unstructured search (complete graph) given by :

$$t = \frac{1}{2\epsilon} \frac{N}{\sqrt{N-1}} \left[ \arctan(\sqrt{N-1}(2s-1)) + \arctan \sqrt{N-1} \right] \quad (2.6)$$

By inverting the function,  $s(t)$  is obtained as plotted in fig. 2.1(a)

The shape of this interpolating schedule comes from the 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 interpolating schedule defined as follow

$$s(t) = \frac{1}{2} (2(x-1)^3 + 1) \quad (2.7)$$

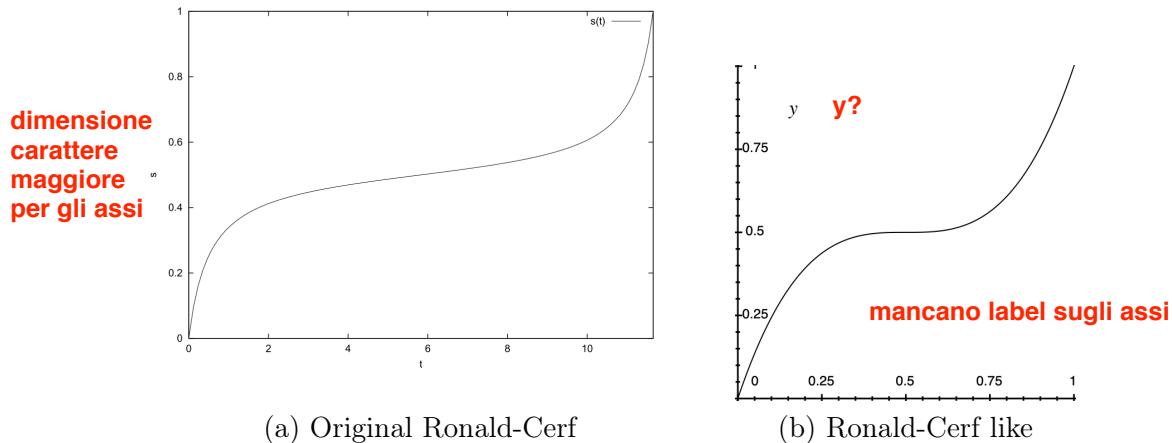


Figure 2.1: **Ronald and Cerf's interpolating schedule (a) and our Ronald-Cerf like schedule (b).** These figures show the difference between the original interpolating schedule derived by Ronald and Cerf for the unstructured search (left) and our Ronald-Cerf like schedule(right). Although quite different we study the effects of this particular non-linear interpolating schedule and compare it with the linear one used by Farhi and Gutmann [2].

### 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 probability 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  $|\langle \psi(t_f) | w \rangle|^2$  is greater than  $1/\text{poly}(N)$  (where  $N$  is the dimension of the graph) [3].

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  $t_{\text{init}}$  to prepare the system in the correct state as well a physical time for the measurement. Therefore, computing multiple searches with small  $t_f$  becomes less efficient than less searches with larger  $t_f$ , where the quantity  $t_{\text{init}}$  makes a lesser contribution to the overall  $t_f$ . Clearly this consideration is particularly relevant for increasing graph size.

## 2.2. Selected Topology: Circular and Complete Graph

Throughout our analysis we will focus ~~the attention~~<sup>topologies</sup> on two selected graph topology, the ring graph (R) and the complete graph (C).

As previously discussed in Section? 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 [1] and the local adiabatic evolution [4], 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 [6], however as we shall later see a merely time-dependent approach might give promising results in terms of robustness.

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

The choice of graph topology is therefore based on the idea of giving results at either side of the spectrum of 2-dimensional graphs, trying to improve a non-working approach on one side, and get some interesting insights on an already perfect-search approach on the other.

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

Firstly it's necessary to define what type of results we're looking for. We begin by showing the difference between **search** and **localization**. Then we introduce a (not-so-rigorous) measure for the **robustness** of the search algorithm.

### 2.3.1 Search vs Localization

Before looking at the results it's necessary to characterize two particular classes of results, the **search** and the **localization**, that help us decide whether the time-dependent approach ~~does~~ 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 **localization** describes the finding of the solution with high probability 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 enough  $t_f$ .

### 2.3.2 Robustness

We've seen that the probability of a search problem depends on the time  $t_f$  at which the quantum state is measured and the parameter  $\gamma$ . The optimal probability, i.e. highest possible, clearly is given by the optimal combination  $t_f - \gamma$ , and since the time  $t_f$  should not be bound to errors give ~~it's~~ **it is** the time at which the state is measured, fluctuations in the probability are necessarily caused by fluctuations in the  $\gamma$  parameter.

**ma in realtà ci  
possono  
essere errori  
sul tempo di  
misurazione.  
Lo strumento  
di misura ha  
una sua  
sensibilità**

Following from [5] we define the **robustness** ~~a~~ quantitative measure representing the variation on the probability due to some perturbation/noise on  $\gamma$ . We begin by finding the optimal probability  $P$ , evaluated with the single or multiple run for one search approach. For the corresponding  $(t_f, \gamma)$  combination we evaluate the robustness as follows:

$$R^\pm = P(t_f, \gamma) - P(t_f, \gamma \pm \delta) \quad \begin{matrix} \text{perchè non la} \\ \text{definiamo con il} \\ \text{modulo?} \end{matrix} \quad (2.8)$$

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

$$R = \frac{R^+ + R^-}{2} \quad (2.9)$$

The quantity  $R$  should be positive, since the  $(t_f, \gamma)$  combination should produce the highest probability possible. We also that the perturbation on the parameter has equal probability of being positive or negative, thus the average ponderates between these two possibilities.

It is necessary however to remember that the value  $R$  does not have any absolute physical significance, but fits well for our specific scenario where the interest is focused on the comparison between two specific approaches. Therefore this quantity will be solely used to characterize a particular approach as **more or less robust**.

girarla in positivo

## 2.4. Results for the Circular Graph

We now address the results for the circular graph. We begin by computing benchmarks for the time-independent hamiltonian approach; this should give us a good starting point for the comparison. Then we consider the time-dependent hamiltonian and evaluate the necessary data. We compare the two approaches, beginning from the Localization, 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 search.

### 2.4.1 Time-Independent Benchmarks

beta?

The first step of the analysis is to compute time-independent benchmarks. We computed the probability over a time-beta grid, with  $T \in [0, N]$  and  $\gamma \in [0, 2.5]$ , where  $N$  is the dimension of the graph considered. An initial run of the probability distribution showed that the probability does not increase with time, thus the need to evaluate for  $t_f$  greater than  $T = N$  proved to be unnecessary. In addition Grover algorithm for the unstructured search (and the complete graph) scales  $\sqrt{N}$  and the Farhi and Gutmann's adiabatic evolution scales like  $N$ , thus focusing on  $T \leq N$  made much more sense.

#### Figure of probability distribution for sampled $\gamma$

Throughout the analysis we will consider graphs up to  $N = 71$ , with only odd dimensions which makes easier oracle placement. Additional considerations and reasoning for the time-gamma grid can be found in Appendix A.

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

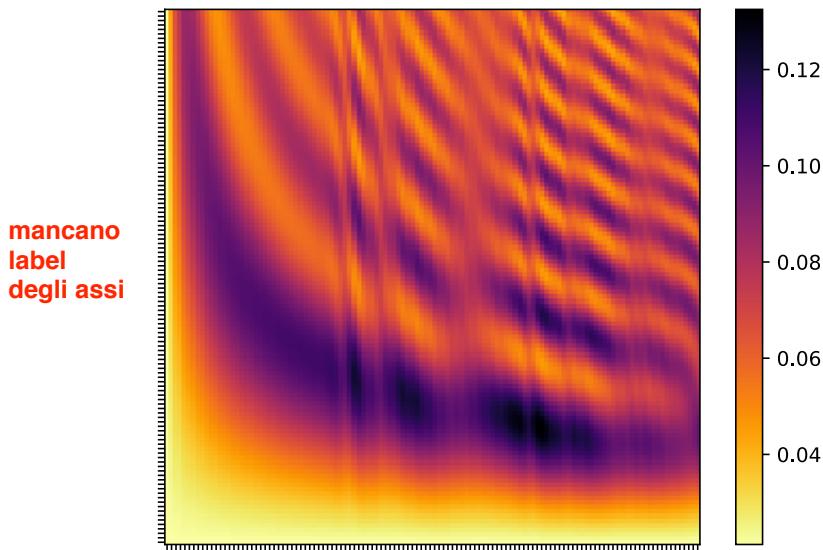


Figure 2.2: **Probability heatmap plot for the time-independent hamiltonian.** The figure shows the probability distribution for a circular graph of  $N=47$  evaluated using the time-independent hamiltonian, providing thus the necessary benchmark. Note that dark color do not represent probabilities close to one, but only higher probability regions.

#### Comments on the probability distribution

From the heatmap plot we see that the probability distribution does not increase smoothly with increasing time, but shows peaks (dark regions) and valleys (lighter regions). Additionally dark regions are scattered throughout the grid and not necessarily for large  $t_f$ . We will later see that this is somewhat a weakness of the time-independent approach, since a small variation of the parameter  $\beta$ , which represents as previously mentioned the deepness of the well or an energy parameter, leads to possibly great variation of the probability.

#### 2.4.2 Time-Dependent Results

Similarly we computed the grid-probability using the time-dependent hamiltonian previously introduced. To easily comparing the two methods we opted to consider the same time  $T = N$ , and from an initial run we discovered that the  $\gamma$  parameter affected similarly the probability (**A plot is needed to support this statement**), namely the probability tended to be higher for smaller values of  $\gamma$ , thus we used approximately the same range.

The grid-probability is evaluated for all the interpolating schedules  $s(t)$  previously discussed, and an intuitive heatmap plot is outputted.

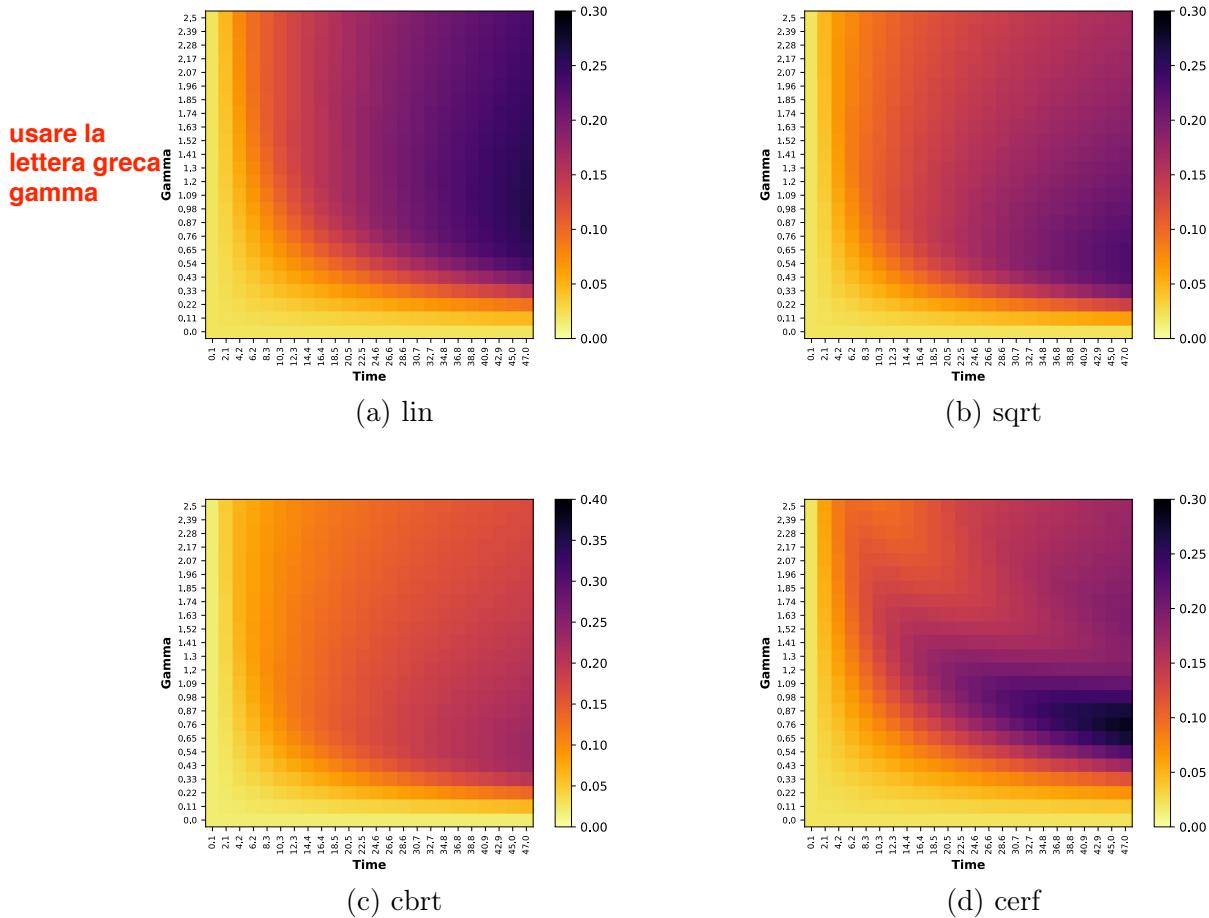


Figure 2.3: **Probability heatmap plot for the time-dependent hamiltonian, for different shapes of  $s(t)$ .** The figure shows the probability distribution for a circular graph of  $N=47$  evaluated using the time-dependent hamiltonian using the following step functions (a) linear, (b)  $\sqrt{t/T}$ , (c)  $\sqrt[3]{t/T}$  and (d) Ronald-Cerf(3). Note that the color gradient is normalized to  $p=0.3$  to accentuate the difference in probability between different regions.

**(C) arriva a 0.4**

### 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) both for constant  $\gamma$  (any horizontal section) and constant  $t_f$  (any vertical section). If we look at the different interpolating schedules **it is** immediately evident that the cbrt and sqrt schedules perform poorly compared to the linear one. On the other hand the Ronald-Cerf schedule performs in a similar way in terms of the highest probability found, but has a significantly different probability distribution, that might affect its robustness.

**distribution**

### 2.4.3 Comparison: Localization

We now compare the localization properties of the two algorithm. As we mentioned in the comments to the probability distributions of the time-independent and time-dependent approaches we discovered the following:

- **Time-Independent QW:** the time-independent based search is not able to solve the search problem with a single iterations, thus 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 in the time-dependent approach.
- **Time-Dependent QW:** the time-dependent based search on the other hand because of the adiabatic theorem does solve the search problem with a single iterations, although that happens for fairly large  $t_f$ . **values of  $t_f$**

Although the time-dependent approach is able to get to unitary probability in a fairly large  $t_f$ , 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  $t_f$  closer it gets to  $P = 1$ . **A plot is needed**

### 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  $t_f$  is not optimized for the time-dependent approach and does not exist for the time-independent one. Therefore, as previously mentioned we consider the possibility of **doing** multiple runs for one search. For this reason we **introduce** the following quantity

$$\min \left( \frac{t_f}{p} \right) \quad \text{a volte parli al passato a volte al presente. Uniformare il tutto} \quad (2.10)$$

where  $1/p$  is the number of iterations necessary to get to unitary probability (statistically), and combining it with  $t_f$  gives the total time necessary. Optimizing over the combination of  $t_f$  and  $p$  gives the smallest time necessary to get to unitary probability using the multiple search approach. Additionally, the number of iterations  $\text{iters} = 1/P$  might give some useful insights on the performance of the approach, in particular if you take into account the initialization time  $t_{init}$ .

However, this particular approach poses a few problems. **In fact** if we look at how the quantity  $t_f/p$  varies for varying  $t_f$ , we discover that the minimum of such quantity will always be for the smallest  $t_f$  available, regardless of the type of hamiltonian and step function  $s(t)$ . The following plot **does** indeed show, for a few sampled  $\gamma$ , the shape of  $t_f/p$ . In addition, in Section ? we mentioned that when dealing with multiple runs for

**minuscolo**

one search we need to take into account an initialization time  $t_{init}$ . It is thus necessary to consider a lower bound on  $t_f$ .

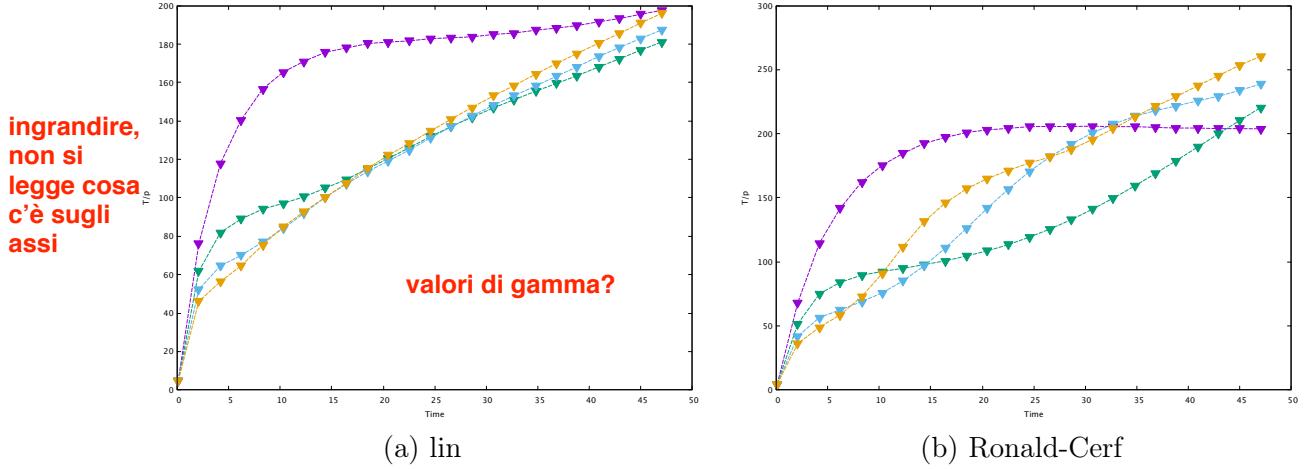


Figure 2.4: Distribution of the quantity  $T/p$  for sampled  $\gamma$  showing that the  $\min(T/p)$  will always be the smallest for the smallest  $t_f$ . The figure shows the distribution of the quantity  $t_f/p$  for some sampled values of the parameter  $\gamma$ , using the linear and Ronald-Cerf interpolating schedules. It is clear that the quantity  $\min(t_f/p)$  will always be for the smallest  $t_f$  available, regardless of the interpolating schedule, requiring therefore a constrain on the time.

### $\min(t_f/p)$ and run iterations for increasing lower bound $t_f$

We will begin by studying how the quantity  $\min(t_f/p)$  and *iters* varies with increasing lower bound  $t_f^*$ . In the following plot we show the shape of  $t_f/p$  with the time-independent approach and the time-dependent one; for the time being and for sake of simplicity, we consider only the step function Ronald-Cerf(3). As we can see from the plot the distribution can be divided into two sections marked by a particular  $T^*$  (for the time being the value of such time is not of our interest):

- for  $t_f < T^*$  the time-independent approach performs slightly better than the time-dependent one.
- for  $t_f > T^*$  however the time-dependent approach performs significantly better, in particular with increasing time  $t_f$

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?, in contrast with the time-independent approach that does not show localization properties.

What this shows is that the choice of  $t_f$  has great effects on the outcome of our time-dependent approach, making it a successful or unsuccessful alternative. In terms of iterations the trend is similar to the quantity  $\min(t_f/P)$ , as the following plot shows.

label più grandi

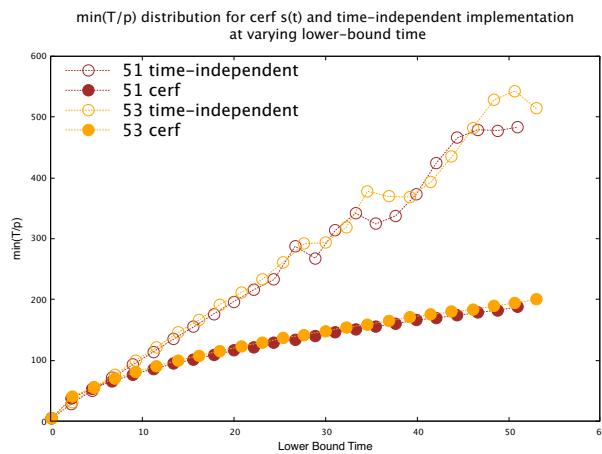


Figure 2.5: **min( $t_f/P$ ) distribution for increasing lower bound time.** The figure shows the distribution of the quantity  $\min(T/p)$  for increasing values of lower bound time, using the time-independent hamiltonian (circles) and time-dependent hamiltonain (solid circles) and evaluated for a Cy(51) and Cy(53). 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.

The iterations distribution 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  $t_f$  increases we see two very different trends:

- The time independent approach requires an almost constant number of iterations (the numerical values is somewhat irrelevant since this distribution reflects only the Cy(53) and Cy(55)). This reflects the non-localization properties of this particular approach, for which the probability does not increase with time
- On the other hand the time-dependent approach, showing localization properties, requires less iterations to solve the search with unitary probability.

Taking into account that we're performing a multiple run search and the initialization time  $t_{init}$  previously discussed, it is clear that the time-depedent approch performs better than the time-independent counter part in most of the scenario.

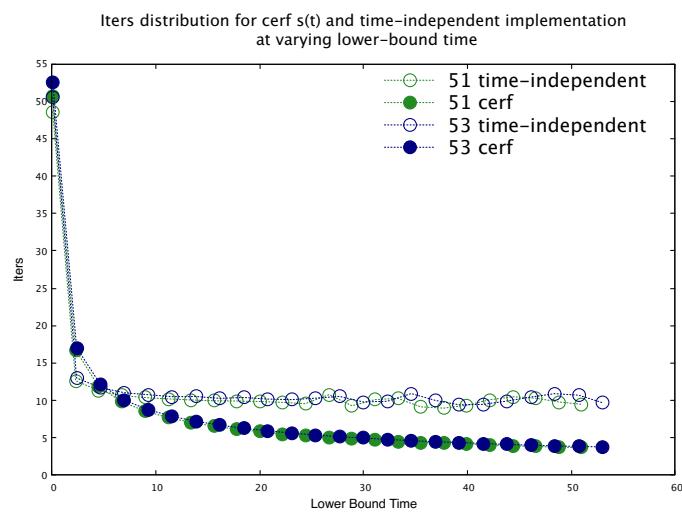


Figure 2.6: **iters distribution for increasing lower bound time.** The figure shows the distribution of *iters* for increasing values of lower bound time, using the time-independent hamiltonian (circles) and time-dependent hamiltonain (solid circles) and evaluated for a Cy(51) and Cy(53). This distribution 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 *iters*, while the time-dependent hamiltonian showing localization properties requires less iterations to get to unitary probability.

### $\min(t_f/p)$ and run iterations with constrained lower bound time

In order to show that the lower bound time does indeed have such great impact on the performance of the time-dependent approach relative to the time-independent one we now study the distribution of the quantity  $\min(t_f/P)$  with a constrain on the lower bound time.

The choice of constrain is arbitrary and somewhat biased since the larger the constrain the better the performance of the time-dependent approach, as we've just shown in fig. 2.7. Therefore, to make the choice fair, we consider the lower bound time to be  $t_f^* = 2\sqrt{N}$  which is (roughly) the time scaling of the standard Grover's Search, the QW Search on the complete graph and so on. In the best case scenario we discover that the number of iterations necessary to get to unitary probability remain constant regardless of the dimension of the graph, making this approach scale as the ones just mentioned; in the most probable scenario we discover that the number of iterations increase with the graph size, thus adding a scaling factor that depends on some power of  $N$ .

The following plot shows the distribution of the quantity  $\min(t_f/P)$  for circular graphs  $Cy(N)$  with  $N$  up to 71. The quantity is computed using the time-dependent hamiltonian with linear step function and the time-independent one, with constrained time  $t_f = 2\sqrt{N}$ .

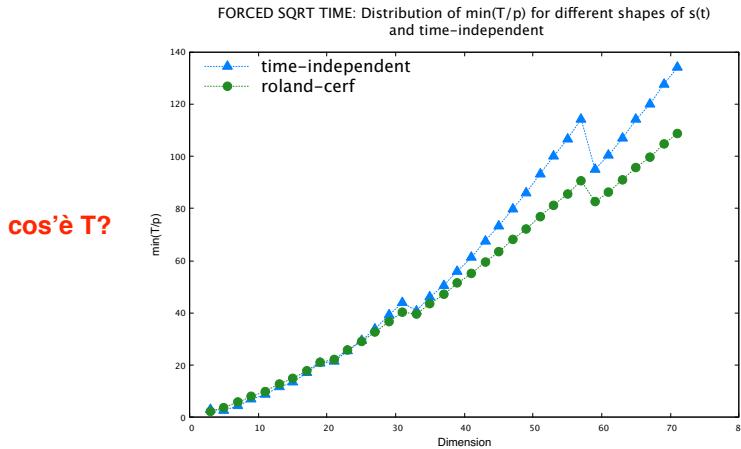
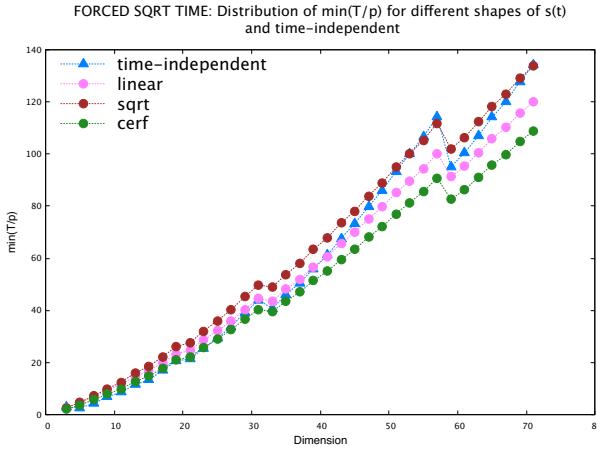


Figure 2.7:  $\min(t_f/P)$  distribution for  $Cy(N)$  up to  $N=71$  with constrained time at  $2\sqrt{N}$ . The figure shows the distribution of the quantity  $\min(T/p)$  for increasing values of lower bound time, using the time-independent hamiltonian (circles) and time-dependent hamiltonian (solid circles) and evaluated for a  $Cy(51)$  and  $Cy(53)$ . 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.

### $\min(t_f/p)$ and run iterations for different shapes of $s(t)$

We now investigate the effects of the different step functions discussed in section? in the probability distribution and in particular on the quantity  $\min(t_f/P)$ . As we did for

the general time-dependent and time-independent comparison we constrain the lower bound time to be larger than  $2\sqrt{N}$ . The following plot shows the distribution of the quantity  $\min(t_f/P)$  for circular graphs  $Cy(N)$  with  $N$  up to 71, using the time-dependent hamiltonian with linear, sqrt, cbt and Ronald-Cerf(3) interpolating schedules  $s(t)$ :



**Figure 2.8:  $\min(t_f/P)$  distribution for  $Cy(N)$  up to  $N=71$  with constrained time at  $2\sqrt{N}$  and different interpolating schedules.** The figure shows the distribution of the quantity  $\min(T/p)$  with constrained lower bound time at  $t_f = 2\sqrt{N}$ , using the time-independent hamiltonian and time-dependent hamiltonain with different interpolating schedules. The abrupt changes are due to the resolution of the grid-probability evaluation (further discussions can be found in Appendix A)

We can see from the plot that, as we might have expected, the Ronald-Cerf interpolating schedule performs better than the linear and sqrt schedules, in particular when we consider larger graph. For small enough  $N$  the time-independent approach performs slightly better, while for large  $N$  the time-depenedent approach is superior. Please note that the abrupt changes in the plot are due to the resolution of the grid-probability evaluation, discussed in Appendix A. *nota: Additional computations will be run to try to fix this issue.*

#### 2.4.5 Comparison: Robustness

We now address the robustness of the time-dependent and time-independent search. As we mentioned in Section? **we are** we're only interested in the comparison of the two approaches, and not an absolute measure of their robustness. Therefore we will use this measure solely as a comparison value.

We proceed by considering small variations on the parameter  $\gamma$  in the order of 1% and 5% (this numbers still need to be discussed). We begin by finding the quantity  $\min(t_f/P)$  with time constrains ( $t_f = 2\sqrt{N}$ ). For the corresponding  $(t_f, \gamma)$  combination we evaluate the robustness  $R$  as defined in Section?. For the time-dependent search with only consider the linear step function an the Ronal-Cerf(3). As we showed in the previous section the Ronald-Cerf Hamiltonian performs better than the linear conterpart, while from a

qualitative point of view the linear hamiltonian has a smoother probability distribution (see fig. 2.3(a)-(d)). The following plots show the semi-quantitative robustness for the  $\gamma$  variations in the order of 1% - 5% for the time-independent search and the time-dependent one with linear and Ronald-Cerf step functions  $s(t)$ . **Computations are still needed**

## 2.5. Results for the Complete Graph

### 2.5.1 Search results from Wong(2016)

### 2.5.2 Localization results

# Appendices

## Appendix A: Probability Grid Evaluation

## Appendix B: Computational Routines

In this section an overview of the computational methods is presented, focusing the attention on the **optimization algorithm** and the **differential equation solver**. Additionally the normalization error is discussed. Lastly computational reasoning for the **probability grid evaluation** are presented.

Most all numerical simulations were performed using **Python**. Numerical methods such as optimization and ODE Solver come directly from python's native **Scipy**. In addition, a CPU-multiprocessing library, **Ray**, has been used to speed up the grid probability evaluation quite noticeably. Heatmap plots were created using python matplotlib, while additional plots were created with gnuplot.

### 2.5.3 Optimization Algorithm

In Section III a series of benchmark were performed to compare the time-dependent and time independent hamiltonian approach to the search problem. In order to determine which optimization algorithm fitted the best for the task, a number of possible algorithm were tested, such as *shgo*, *dualannealing*, *minimize*, *LHSBH* and *Basin-Hopping*.

Due to the oscillating nature of the probability (a figure is needed) the scipy native **Basinhopping algorithm** was used. As the name suggests the algorithm performs a series of randomized hops, i.e. jumps, of the coordinates in order to find the true maximum. This fits particularly well with the series of maxima and minima of the probability function (for fixed  $\gamma$ ) in the time-independent hamiltonian.

Additional information on the parameter used are needed (e.g. step size, number of iterations)

#### 2.5.4 Schroedinger Solver and Normalization Error

In Section II we presented an evolution which is governed by a time-dependent hamiltonian, used to find the evolved state  $|\psi(t)\rangle$ . This is accomplished by solving the usual Schrödinger equation using Scipy's `integrate.solve_ivp`, that provides a wide varieties of integrations methods.

As it's routine we used Runge-Kutta RK45, which as stated in the documentation it's an explicit Runge-Kutta method of order 4(5). The error is controlled assuming fourth order accuracy, but steps are taken using the fifth-order accurate formula. In addition, the integrator is adaptive, meaning that the time step is chosen for optimal error control. Regarding the error, the algorithm provides two distinct parameters to set a targeted limit, namely the **relative (rtol)** and **absolute tolerances (atol)**. The first provides a relative accuracy, i.e. the number of digits, while the latter is used to keep the local error estimate below the threshold  $atol + rtol*abs(y)$ . Determining the correct combinations of the two parameters is key for achieving the desired error. A few of those are presented in the following table, where a worst case scenario ( $N=101$ ,  $T=1000$ ) is used and the error is evaluated on the expected normalized state. The combination of **rtol** and **atol** bolded is the one we used in the solver, which gives us a small enough error on the normalization without greater computational expense.

# Bibliography

- [1] A. M. Childs and J. Goldstone. Spatial search by quantum walk. *Physical Review A - Atomic, Molecular, and Optical Physics*, 70(2), 2004.
- [2] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. Quantum Computation by Adiabatic Evolution. 2000.
- [3] J. G. Morley, N. Chancellor, S. Bose, and V. Kendon. Quantum search with hybrid adiabatic?quantum walk algorithms and realistic noise. pages 1–24, 2018.
- [4] J. Roland and N. J. Cerf. Quantum search by local adiabatic evolution. *Physical Review A - Atomic, Molecular, and Optical Physics*, 65(4):6, 2002.
- [5] S. Z. SH. Hung, S. Hietala. Quantitative Robustness Analysis of Quantum Programs (Extended Version). *Proceedings of the ACM on Programming Languages*, 3(January), 2019.
- [6] T. G. Wong and D. A. Meyer. Irreconcilable difference between quantum walks and adiabatic quantum computing. *Physical Review A*, 93(6):1–8, 2016.