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

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 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

List of Figures

Introduction

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:

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

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

$$L = A - D \tag{1.2}$$

which we'll later see describes the quantum walk evolution.
In the quantum realm a vertex is a vector in an N dimensional Hilber space, so that a vertex $|j\rangle$ can be represented in the following way

$$|j\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ .. \\ 0 \end{pmatrix} \tag{1.3}$$

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

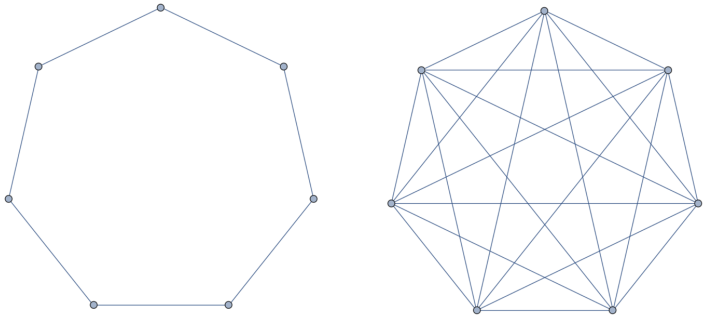


Figure 1.1: Pictorial representation of a circular graph (left) and a complete graph (right) with N=7

1.2. Quantum Walks

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

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

where $p_j(t)$ is the probability of being on the vertex j at time t.
The quantum analogue takes place in an N-dimentional Hilbert space spanned by the states (vertex) $|j\rangle$ of the graph G. Instead of considering the classical probability as we previosuly 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 form of

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

The continous-time quantum walk is defined by letting $H = -\gamma L$, where L is the previosuly defined Laplacian matrix.

1.3. Grover's Quantum Search

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

1.4. Quantum Search with Continous-Time Quantum Walk

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

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

$$H_w = -|w\rangle\langle w| \tag{1.6}$$

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

$$H = -\gamma L + H_w = -\gamma L - |w\rangle\langle w| \tag{1.7}$$

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.

The quantum search routine works as follow:

- we consider the superposition of all possible states, namely

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_j |j\rangle \tag{1.8}$$

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

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

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

- we then measure the state onto the target $|w\rangle$ and find the corrisponding probability

$$p = |\langle w|\psi(T)\rangle|^2 \tag{1.10}$$

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

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**

A quantum system evolves according to the Schroedinger equation

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

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

$$H(t)|l;t\rangle = E_l(t)|l;t\rangle \tag{1.12}$$

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

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

This means that if the system is chosen to evolve at a slow enough rate, the instantaneous hamiltonian will remain in the ground state throught 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

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

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

$$T \gg \frac{\varepsilon}{g_{min}^2} \tag{1.15}$$

where

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

1.5.1 Computation by Adiabatic Evolution

Let's now discuss how to take advantage of the adiabatic theorem introducting 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 abeginning 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

$$H(s) = (1 - s)H_B + sH_P \tag{1.17}$$

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 ourself 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

We now turn our attentiont to a time-dependent implementation of the hamiltonian governing the evolution of the quantum walks. In particular we look at the application of such QW to the search problem on two selected graph topology, the complete and the circular graph. These two represent respectively the best and worst case scenario, namely the complete graph is known to work with the time-independent implementation [1] and the local adiabatic evolution [4] from which we constructed our work.

2.1. Search with Time-Dependent Hamiltonian

2.1.1 Time-Dependent Quantum Walk

Following the adiabatic evolution discussed in the preliminaries, we consider a search hamiltonian consisting in an oracle term and a laplacian term, such that at the beginning of the evolution we hamiltonian is equal to the laplacian, while at the end it's equal to the oracle hamiltonian. Such hamiltonian is constructed in the following way:

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

where $|w\rangle$ represents the target of the search. The function $s(t)$ (referred as *step function*) regulates the evolution of the hamiltonian an plays a crucial role in the final search probability, as we will later see. It is defined in the following way

$$s : [0, T] \to [0, 1] \tag{2.2}$$

where T is the time at which the solution is found (or actually, the time that the system is made to evolve. A solution might not be found at such time, or at least found with

low probability).

2.1.2 Comments on the shape of s(t)

As we mentioned previously the step function s(t) plays a crucial role in the evolution of the system. The classical adiabatic evolution by Farhi et al. [2] consists in a linear step function s(t) defined as $s(t) = \frac{t}{T}$, such that the systems evolves linearly.

We however investigate which is the optimal shape of s(t) for otpimal probability. We first consider, somewhat naively, the square root and the cubic root of t/T, and the square and third power of t/T. However, for the complete graph Ronald & Cerf propose that the optimal shape of s(t) comes from considerations on the separation of the first two eigenvalues [4]; in particular s(t) should be steeper for larger separation and flatter for small eigenvalues separation. This in indeed somewhat to be expected, considering that for the adiabatic theorem to hold the adiabatic time needs to have a certain value which dependes directly from the eigenvalues separation.
Following that approach we consider a similar step function that tries to emulate the one from Ronald & Cerf. For this paricular reason we will refer to this particular step function as the Ronald-Cerf s(t) even though it's not exacly the same one they derived.

Follows here a summary of all the step functions considered in the analysis of the circular graph. Note that for the complete graph, since the solution is derived analitically such different s(t) are not considered nor of particular interest.

- **lin**: $s(t) = \frac{t}{T}$
- **sqrt**: $s(t) = \sqrt{\frac{t}{T}}$
- **Ronald-Cerf(3)**: $s(t) = \frac{1}{2}(2(x-1)^3 + 1)^2$
- **Ronald-Cerf(5)**: $s(t) = \frac{1}{2}(2(x-1)^5 + 1)^2$

For the Ronald-Cerf step function we considered different values of odd-k, $k = 3, 5, 7$. It is clear that we need to include a picture of the different functions considered and the eigenvalues separation picture for a particular case.

2.1.3 Methods and Dynamics

The time evolution operator follows from the time-ordering approach, namely:

$$S(t, t_0) = Texp\frac{1}{i\hbar}\Big\{ \int_{t_0}^{t} dt' H(t') \Big\} \tag{2.3}$$

where T is the time-ordering.
In order to find the evolved state we can proceed in the following ways:

- **Solve the Schroedinger differential equation**

Since we're only interested in finding the evolved state, having the exact form of the evolution operator is somewhat irrelevant. Focusing our attention on the solution of the differential equations also helps us from a computational point of view; indeed the implementation of the differential equation solver is much simpler than any of the other two possibilities.

Solving the Schroedinger Equation

From the intial state $|\psi\rangle$ we're intested in finding the evolved state $|\psi(t)\rangle$, and we proceed solving the differential Schroedinger equation:

$$i\frac{d}{dt}|\psi(t)\rangle = H|\psi(t)\rangle \tag{2.4}$$

Since H is a matrix we had to solve it for components. Recalling that $|\psi\rangle$ is a vector in an N-dimensional Hilbert space, we need to solve N-differential equations of the form:

$$\frac{d}{dt}|\psi_i(t)\rangle = \sum_j H_{ij}|\psi_i(t)\rangle \tag{2.5}$$

We also need to impose boundary conditions, namely that $|\psi(0)\rangle = |\psi\rangle$, our initial ground state.

2.1.4 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 $\left|\langle\psi(t_f)|w\rangle\right|^2$ is greater than $1/poly(N)$ (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_{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_{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 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 and the local adiabatic evolution, with a speed up of $\sqrt{N}$. An adiabatic implementation -where with adiabatic we consider the time-dependent approach with additional adiabatic-theorem conditions - of the quantum walk search does not work with the usual Grover's oracle requiring a more elaborate structure, however as we shall later see a merely time-dependent approach might be somewhat interesting.

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

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 bring 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.
- the **localization** describes the finding of the solution with high probablity without the need to optimize the time. This fenomenological 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 γ. 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 the time at which the state is measured, fluctuations in the probability are necessarily caused by fluctuations in the γ parameter.
Following from [5] we define the **robustness** as a qualitative measure representing the variation on the probability due to the variation on γ. Since our work focuses on the comparison between the time-independent and time-dependent approach, and thus we're interesting in improving on the first with the latter, the use of this term will be by describing a particular approach to be **more robust** or **less robust**. Additionally we could refer to the approach to **show (non)-robustness properties**.

How do we measure robustness? To quantitatively measure robustness we begin by determining the combination of $t_f - \gamma$ that solves the search problem (in a single or multiple run), i.e. the probability is maximum. We then make small changes of γ in the order of 1 to 10 percent, and measure the difference of probability produced:

$$P[t_f, \gamma] - P[t_f, \gamma^*] = \text{some-quantity} \tag{2.6}$$

where γ^* is the *perturbed* parameter.

2.4. Results for the Circular Graph

2.4.1 Time-Independent Benchmarks

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 $\beta \in [0, 2.5]$, where N is the dimension of the graph considered. An initial run of the probability distribution showed that *it is worse than linear in T*, thus the need to see times greater than $T = N$ proved to be unnecessary. In addition Grover algorithm for the unstructured search (and the complete graph) goes like $\sqrt{N}$ and the adiabatic (globally, not locally) goes like N, thus focusing on $T \leq N$ made much more sense. **It's interesting to show a figure with the probability distribution for some sampled β even for large time, giving the idea that the probability does indeed not increase with time and we can focus the attention to smaller T (T<N)**. Throughout the analysis we will consider graphs up to N = 71, with only odd dimensions which makes a easier oracle placement (hamiltonian is somewhat symmetric). Additional considerations and reasoning for the time-beta 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, β):

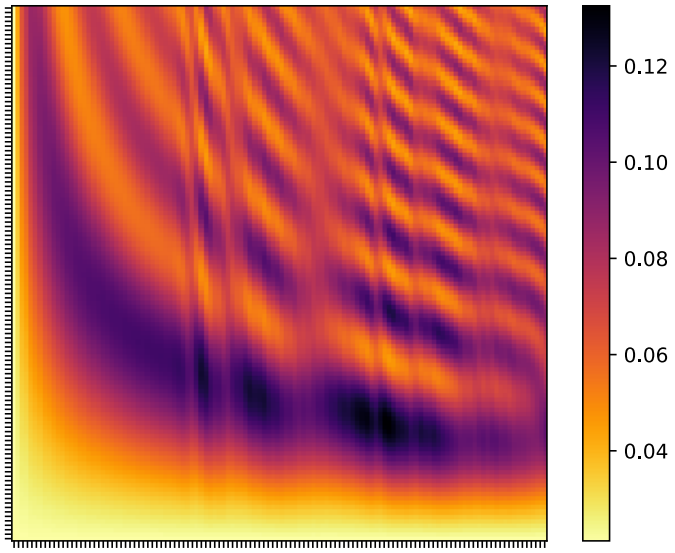


Figure 2.1: **Probability heatmap plot for the time-independent hamiltonian.** The figure shows the probability distribution for a circular graph of N=47 simulated using the time-independent hamiltonian, providing thus the necessary benchmark. Note that darker color do not represent probabilities close to one, but only higher probability regions.

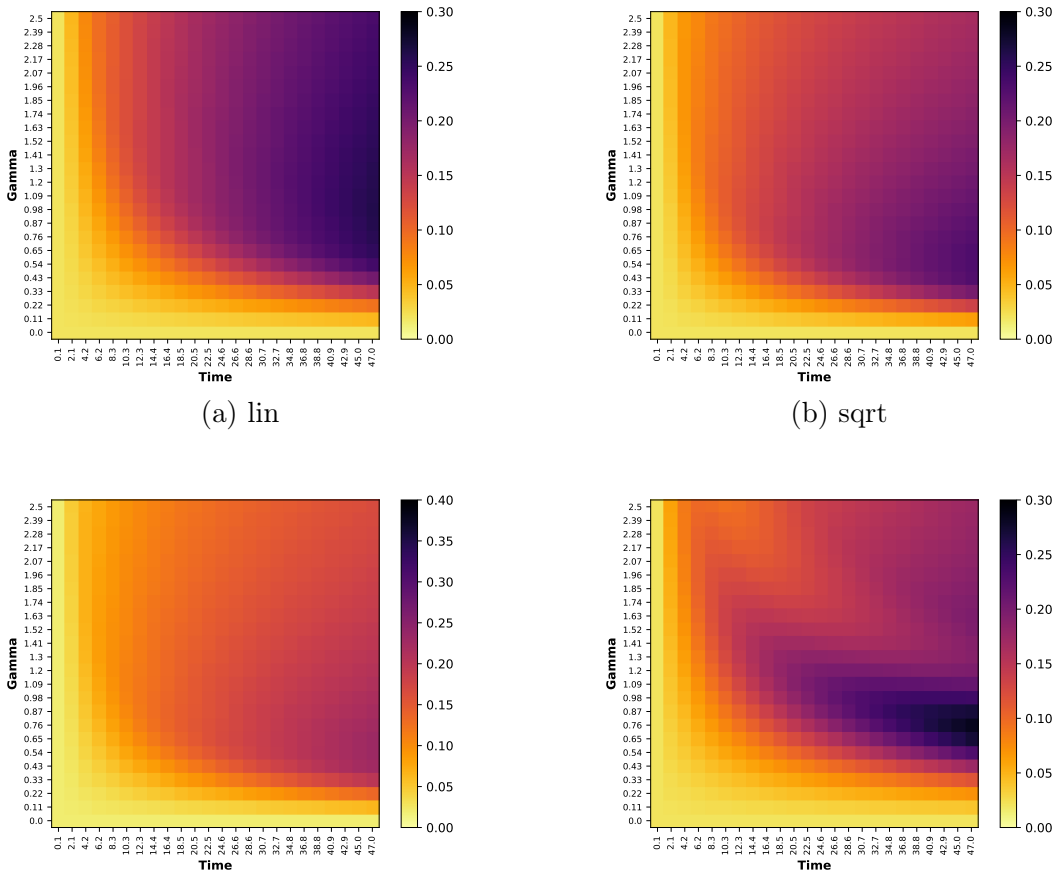
Comments on the probability distribution

It is quickly noticeable that the probability distribution is not smooth, indeed it shows peaks (darker regions) and valleys (lighter regions). This is both present for fixed β, i.e. any orizontal line, and for fixed time, i.e. any vertical section considered. We will later see that this is somewhat a weakness of the time-independent approach, since a small variation of the parameter β, which reperesents as previosly mentioned the deepness of the well or an energy parameter, leads to possibly great variation of the probability. We shall call this fenomena **non-robustness**.

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 γ parameter affected similarly the probability, namely the probability tended to be higher for smaller values of γ, thus we used approximately the same range.

The simulation is then run for all the step functions $s(t)$ previsouly discussed, and an intuitive heatmap plot is outputted.



(a) lin (b) sqrt

(c) cbrt (d) cerf

Figure 2.2: **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 probability gradient is normalized to p=0.3 to accentuate the difference in probability between different regions.

Comments on the probability 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.

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 $P = 1$. **include plot**

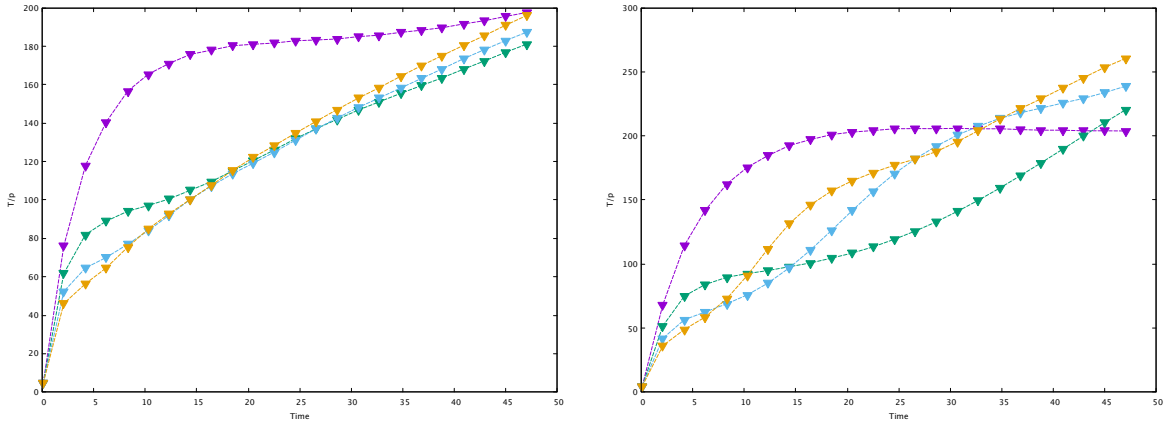
2.4.4 Comparison: Search

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

$$\min\left(\frac{t_f}{p}\right) \tag{2.7}$$

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.

However, this particualar approach poses a few problems. Infact 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 γ, the shape of t_f/p. In addition, in Section ? we mentioned that when dealing with multiple runs for 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.



(a) lin (b) Ronald-Cerf

Figure 2.3: **Sampled T/p to show that the min(T/p) will always be the smallest T available.**

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) varies with increasing lower bound t_f^*. Additionally the number of iterations $iters = 1/p$ represent an interesting quantity, since it's clear that the least number of runs are necessary the better. 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 semplicity, we consider only the step function Ronald-Cerf(3).

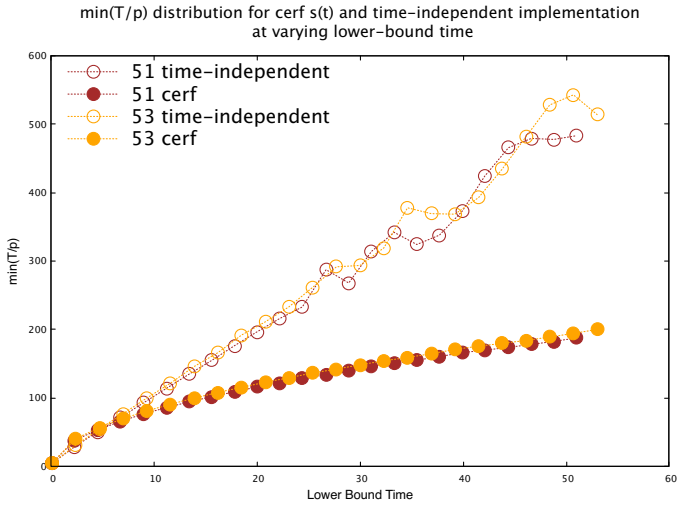


Figure 2.4: **Sampled T/p to show that the min(T/p) will always be the smallest T available.**

As we can see from the plot the distribution can be diveded into to section 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 performes slightly better than the time-dependent one. In terms of run iterations the two approaches show very similar results.
- 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 shoed in Figure?, in contrast with the time-independent approach that does not show localization propeties.
What this show is that the choice of t_f has great effects on the outcome of our time-dependent approach, making it a successfull or unsuccesfull alternative. 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. In order to determine whether this statement is true, we now study the distribution of min(t/p) for multiple graphs, with the lower bound constrain fixed to certain values.

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

2.4.5 Comparison: Robustness

We now address with a semi-quantitative approach the robustness of the time-dependent and time-independent search. Since we're only interested in the comparison of the two approaches, and not an absolute measure of their robustness, as mentioned in Section? we study the difference of probability for set perturbance of the γ parameter. In particular we investigate for fairly small variation on the parameter (1%), up to quite large variations (5% and 10%).

We begin by finding the quantity $\min(t_f/P)$ with time constrains ($t_f = \sqrt{N}$). For the corresponding (t_f, γ) combination we evaluate the robustness as follows:

$$R^{\pm} = P(t_f, \gamma) - P(t_f, \gamma \pm \delta\gamma) \tag{2.8}$$

where $\delta\gamma$ is the variation of the parameter (1, 5 or 10% of γ), that can be both positive and negative. To find a unique value for the robustness an average of $R^{\pm}$ is done:

$$R = \frac{R^+ + R^-}{2} \tag{2.9}$$

In theory the quantity R should be positive, since the (t_f, γ) combination should produce the highest probability possible. The average over positive and negative variations ensure that each approach are treated fairly.

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. The following plots show the semi-quantitative robustness for the γ variations in the order of 1% - 5% - 10% for the time-independent search and the time-dependent one with linear and Ronald-Cerf step functions s(t). **Insert figure**

2.5. Results for the Complete Graph

2.5.1 Search results from Wong(2016)

2.5.2 Localization results

Conclusions

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 γ) 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 Schroedinger 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 a 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.