

# Systematic comparison between time-independent and time-dependent hamiltonian approach to quantum search

Matteo Garbellini  
Department of Physics  
Università degli Studi di Milano

(Dated: July 18, 2020)

Keywords: Quantum Search, Quantum Walk, Adiabatic Theorem

Legend:

Red text is comment text with general guidelines on what to write in that particular section.

Blue text is for figures and tables needed.

Green text is additional equation needed.

Purple text is for references.

## I. INTRODUCTION

This report can be summarized as follows. In Section II a review of the background topics is presented, with particular emphasis on Quantum Walks, Quantum Search and the Adiabatic Theorem. A brief introduction on Graph Theory is presented as well, in order to define the working context. In Section III we discuss the Dynamics, focusing on the time dependent and time independent hamiltonian approach to quantum search, as well as the Methods, i.e. the numerical solving of schroedinger equation and more broadly the numerical evolution of the system. A brief but not less important introduction to the computational methods (e.g. optimization algorithms and solver) is presented. In Section IV we present results for selected topologies, namely the *worst-case-scenario* Circular Graph and the *best-case-scenario* Complete Graph. For the two selected topologies we compare the known time-independent approach and the time-dependent, in particular defining two qualitative characteristics for results: optimized search and localization. Section V presents a summary of the work done so far, and the further developments to be made.

## II. BACKGROUND

### A. Graph Theory

Basic introduction to the adjacency matrix, diagonal (degree) matrix. Some information on the two topologies considered, namely the circular graph and the complete graph. Once introduced in particular the laplacian matrix there is no need to talk about them in detail.

Picture of circular and complete graph

maybe an example laplacian/adjacency matrix for a small N

### B. Quantum Walks

What is a quantum walk starting (maybe) from the notion of the classical random walk. Some words on discrete-time and continuous-time quantum walks and the fact that we consider the latter. Define the dynamics of the walker in terms of the laplacian matrix.

### C. Quantum Search with QWs

Introduction to quantum search with the classical Grover algorithm and how it is possible to formulate the problem with continuous-time quantum walks. This section should contain the main core of quantum search and all the theoretical principles used throughout the thesis. It should be clear what finding the oracle is and what is the probability we're looking for. Description of the system evolution (matrix element for probability) will be discussed in later sections.

### D. Adiabatic Theorem

Key aspect of the adiabatic theorem is the fact that we can get very close to probability  $p=1$  for large  $T$ . Also the beginning-problem hamiltonian feature is quite important to explain with clarity.

A quantum system evolves according to the Schroedinger equation

$$i \frac{d}{dt} |\psi(t)\rangle = H(t) |\psi(t)\rangle \quad (1)$$

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

$$H(t) |l; t\rangle = E_l(t) |l; t\rangle \quad (2)$$

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

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

$$\lim_{T \rightarrow \infty} |\langle l=0; t=T | \psi(T) \rangle| = 1 \quad (3)$$

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

$$g_{min} = \min_{0 \leq s \leq 1} (E_1(s) - E_0(s)) \quad (4)$$

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

$$T \gg \frac{\varepsilon}{g_{min}^2} \quad (5)$$

where

$$\varepsilon = \max_{0 \leq s \leq 1} \left| \left\langle l=1; s \left| \frac{dH(s)}{ds} \right| l=0; s \right\rangle \right| \quad (6)$$

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

$$H(s) = (1-s)H_B + sH_P \quad (7)$$

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.

### III. DYNAMICS AND METHODS

In this section we look more in depth at the time-independent and time-dependent hamiltonian used for the search algorithm, putting particular focus on the time function that regulates the evolution of the hamiltonian. We then explore the possible way to numerically solve the schroedinger equation, giving reasoning of why we decided to solve it numerically using a Runge-Kutta integrator. Last but not least we look at the computational methods, namely the code written and the optimization and integrator algorithms used.

### A. Time Independent Hamiltonian

As we've seen in (Section II - Quantum Search), the time independent schroedinger equation is quite easily solved, since the evolution operator  $S(t)$  is defined as follows:

$$S(t) = e^{-\frac{i}{\hbar}Ht} \quad (8)$$

and therefore, given a state  $|\psi\rangle$  its evolution is given by

$$|\psi(t)\rangle = S(t)|\psi\rangle = e^{-\frac{i}{\hbar}Ht}|\psi\rangle \quad (9)$$

### B. Time Dependent Hamiltonian

We introduce a time dependent hamiltonian that follows from the adiabatic implementation to problem solving and the regular time-independent approach previously discussed. The intuition behind this implementation comes from the idea of *adiabatically* turn on the oracle hamiltonian and turn off the laplacian of the graph. Theoretically this should give us the advantage discussed in (Section II - Adiabatic Theorem), namely the high probability of finding the solution with the possible expense of longer time scaling.

The hamiltonian is thus in the form of

$$H(t) = (1 - s(t))L + \gamma s(t)|w\rangle\langle w| \quad (10)$$

where  $s(t)$  is a function of time that will be later discussed in more detail.

### C. Comments on the form of $s(t)$

Discussion on the time-stepping function  $s(t)$  (or better  $g_T(t)$ ). Why did we choose this particular shape for the function? Are we expecting any improvements?

In particular if we follow what has been said by Cerf et al. the function should be steeper (evolving faster) when the separation  $E_1(t) - E_0(t)$  is larger, and slower for small separation. Looking at the separation distribution the functions used (sqrt, cbtr, linear) do not fit to this model.

It's interesting to show, probably in the Results section rather than here, how significant is the shape of  $s(t)$  in the final probability

### D. Computational Methods

In the previews two subsection we discussed time dependent and independent hamiltonians and how these are solved to find the desired probability. In the time-independent case the calculation is straight forward, while in the time-dependent it's necessary to solve numerically Schroedinger equation. In this subsection, called indeed computational methods, a discussion on what algorithms where used to optimize the probability, solve the Schroedinger equation and the probability grid-evaluation is due. Additional information on the specific parameters (e.g. rtol-atol in RK45, basinhopping jumps and shgo iterations) should be presented in the appendix.

## IV. RESULTS FOR SELECTED TOPOLOGIES

### A. Search vs Localization

It's necessary to distinguish between two categories of results: optimized search and localization. This gives an idea on whether the found results are a win or a lost. They two approaches are not exclusive

We shall describe the possible outcomes of the previously introduced hamiltonian and characterize them as **localization** and **optimized search**. The first describes the finding with high probability of the solution without particular interest in the time needed to reach that particular solution. For example, as we shall see later in Section V, the ring-graph is known for not working with traditional quantum-walk algorithm (i.e. found with low probability),

but thanks to the adiabatic nature for  $T \rightarrow \infty$  the solution is found with probability 1.

On the other hand we know that for our algorithm to be somewhat interesting from the perspective of the quantum search we need at least the same speedup we get from the traditional quantum-walk Grover, namely  $O(\sqrt{N})$ . We're thus optimizing for maximum probability and minimum time; in addition as we shall see later we'll also consider the possibility of repeating the search an  $n$  amount of times, and get a less than optimal search for each iteration but better time (sum of time, actually) overall. It is indeed this what we call *optimized search*.

### B. Circular graph

This section should include a comparison between time independent and time dependent approach. The time-independent results come from the initial benchmarks, while time-dependent comes from grid-evaluation. To compare the two methods the quantity  $\delta = \min(T/p)$ . Noticing that the optimized search does indeed not work, or at least is not comparable to the time-independent classical-qw-search, the attention is focused on the high probability found (with large times however), namely the localization property of the algorithm

### C. Complete graph

This section should include a comparison between time dependent and time independent approach to quantum search for the complete graph. Since this topology is solvable analitically I wonder what kind of results should I put in this section.

It might be interesting looking into the order of the probability for  $p=0.9$ . Analytically the solution is looked for  $p=1$ , possibly (for localization in particular) we could get some speed up over  $\sqrt{N}$  for probabilities less than 1. This might work particularly well if the probability is not linear in the vicinity of  $\sqrt{N}$  Probability distribution for fixed  $\gamma$  with increasing time

## V. APPENDIX

In this section an overview of the computational methods is presented, focusing the attention on specific matters: *Optimization Algorithm* used for determining the optimal  $\gamma$ -configuration for maximum probability, *Schroedinger Solver* for the time dependent hamiltonian evolution and some additional parameters for the optimal *Runge-Kutta Normalization Error*. Lastly computational reasoning for the *2D - Probability Heatmap* are presented.

Most all numerical simulations were performed using **Python** being easy and flexible to sudden changes. 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. 2D - Heatmap plots were created using python matplotlib, while additional plots were created with gnuplot.

### A. Optimization Algorithm

In Section II a series of benchmark were performed to compare the standard Quantum-Walk algorithm to the Adiabatic-Quantum-Walk implementation (later simply named *dynamic*). 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 (cfr figure) 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 (actually, the true *minimum*). This fits particularly well with the series of maxima and minima of the probability function (for fixed  $\gamma$ ) in the static algorithm (std QWAlgorithm). Snippets of the parameters used follow.

```

1 from scipy.optimize import basinhopping
2 optimization = optimize.basinhopping()
3 print(optimization)

```

## B. Schroedinger Solver

In Section III 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 is used and the error is evaluated on the expected normalized state.

rtol	atol	norm	error	comp time (s)
e-3	e-5	1.0851	-0.0851	83.8
<b>e-3</b>	<b>e-6</b>	<b>1.0198</b>	<b>-0.0198</b>	<b>19.5</b>
e-3	e-7	1.0188	-0.0188	19.4
e-3	e-8	1.0187	-0.0187	19.5
e-4	e-5	0.9988	1.16e-3	90.6
e-4	e-6	0.9996	3.98e-4	21.7
e-4	e-7	0.9996	4.13e-4	21.4
e-4	e-8	0.9996	4.04e-4	21.5
e-5	e-5	0.9989	1.09e-3	82.8
e-5	e-6	0.9998	1.81e-4	20.5
e-5	e-7	0.9999	1.33e-4	20.6
e-5	e-8	0.9999	1.21e-4	20.8
e-6	e-5	0.9998	2.48e-4	92
e-6	e-6	0.9999	5.78e-5	22.6
e-6	e-7	0.9999	2.47e-5	23.2
e-6	e-8	0.9999	1.68e-5	23.2

## C. Comments on heatmap and computational time

## D. Runge-Kutta normalization errors and parameters