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 20, 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.
figure: Blue text is for figures and tables needed.
equation: Green text is additional equation needed.
reference: 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 enphasis 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.
figure: Picture of circular and complete graph
equation: maybe an example laplacian/adjacency matrix for a small N

B. 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

$$\frac{d}{dt}p_j(t) = \gamma \sum_k L_{jk} p_k(t) \tag{1}$$

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

$$i\frac{d}{dt}q_j(t) = \sum_k H_{jk}q_k(t) \tag{2}$$

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

C. Quantum Search with Continous-Time Quantum Walk

reference: 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.

Grover problem without 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

$$H_w = -|w\rangle\langle w| \tag{3}$$

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

$$H = -\gamma L + H_w = -\gamma L - |w\rangle\langle w| \tag{4}$$

where L is the laplacian of the graph, which contains the information of the dynamics over that particular graph topology.

We then begin considering the superposition of all possible state, namely

$$|s\rangle = \frac{1}{\sqrt{N}}\sum_j |j\rangle \tag{5}$$

and run the quantum walk for time T. The evolved state is then measured in the vertex basis. Our objective is to find the optimal value of γ so that the success probability is as close as possible to 1 for the smallest T

$$p = |\langle w|\psi(T)\rangle|^2 = |\langle w|\psi(T)\rangle| \tag{6}$$

D. Adiabatic Theorem

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

$$i\frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle \tag{7}$$

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

$$H(t)|l;t\rangle = E_l(t)|l;t\rangle \tag{8}$$

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 strictcly greater than zero then for $T \to \infty$ the probability of being in the ground state is equal to one, namely

$$\lim_{T\to\infty} |\langle l = 0; t = T|\psi(T)\rangle| = 1 \tag{9}$$

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

$$g_{min} = \min_{0\leq s\leq 1} (E_1(s) - E_0(s)) \tag{10}$$

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

$$T \gg \frac{\varepsilon}{g_{min}^2} \tag{11}$$

where

$$\varepsilon = \max_{0\leq s\leq 1} \left|\left\langle l = 1; s\left|\frac{dH(s)}{dt}\right|l = 0; s\right\rangle\right| \tag{12}$$

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

$$H(s) = (1 - s)H_B + sH_P \tag{13}$$

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.

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 definded as follows:

$$S(t) = e^{-\frac{i}{\hbar}Ht} \tag{14}$$

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

$$|\psi(t)\rangle = S(t)|\psi\rangle = e^{-\frac{i}{\hbar}Ht}|\psi\rangle \tag{15}$$

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

$$H(t) = (1 - s(t))L + \gamma s(t)|w\rangle\langle w| \tag{16}$$

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, cbrt, linear) do not fit to this model.
It's interesting to show, probably in the Results section rathen 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.

1. *Schroedinger Solver*

2. *Probability Optimization Algorithm*

3. *Computational Routines*

It also might be useful to define some computational procedures so that we can easily refer to them without recalling.
Throught the analysis the goal was to find the optimal probability for the specific scenario, and the procedure varied depending on the graph topology and the hamiltonian considered. That being said, we can characterize them as follows:

- **β and T probability optimization:** as it might imply, we run an optimization using β and time as parameters, trying to maximize the probability. This process is quite computationally intensive, so except for the time-independent algorithm we employ the following two technique.
- **T optimization for fixed β:** this technic is quite useful when we are only qualitatively interested at the probability distribution. Thus sampling a fix number of *beta* and optimizing on the time variable. This results in a great computational time improvement, particularly for the time-dependent approach.
- **probability grid evaluation:** this technique consists in sampling both time and β and evaluating the probability for all the time-beta combinations, namely evaluating the probability on a grid. The advantage of such approach is the much less computational time needed since a single evaluation is much faster than an optimization **figure: might be interesting having a graph comparing computational time**. Secondly true

optimization tries to find the absolute maxima, to the expence of having a larger T (remember, we're trying to find the highest probability for the shortest time). As an example, we might find a probability p=0.95 for T = 10 using the grid evaluation, although the true maximum is p=0.96 for T = 35. Is indeed clear that the 0.01 in greater probability does not balances the disadvatange of more than double time.

IV. RESULTS FOR SELECTED TOPOLOGIES

A. Search vs Localization

It's necessary to distingiush 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 $T \to \infty$ 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

1. *Time-Independent Benchmarks*

The first step in our analysis is to compute some benchmark for the time-independent hamiltonian, in order to later compare the time-dependent approach. This is particularly interesting since we know that th quantum search time-independent algorithm does not work with the circular graph, namely the found probability is periodical and for large N converges to p = 0.1.
figure: Converging probability to 0.1 for large N

Using the time-independent hamiltonian introduced before we optimized the probability on T and β for circular graphs up to N=31. It is worth noting that we considered only odd graphs, since it made an easier center-placement of the oracle site state. The following are the results:
figure: time-independent benchmarks up to N=31

It is clear that the time-independent algorithm does indeed not work, as we expected. The probability decreases as the number of sites N increases, converging for large N to $p = 0.1$.

2. *Time-Dependent Algorithm*

We then considered the time-dependent hamiltonian introduced in Section ?, and following the *probability grid evaluation* procedure discussed before we computed the probability. Given the computational efficency over the regular β-T optimization we were able to compute results up to N = 71, although from 31 to 71 only the 1st and 7th

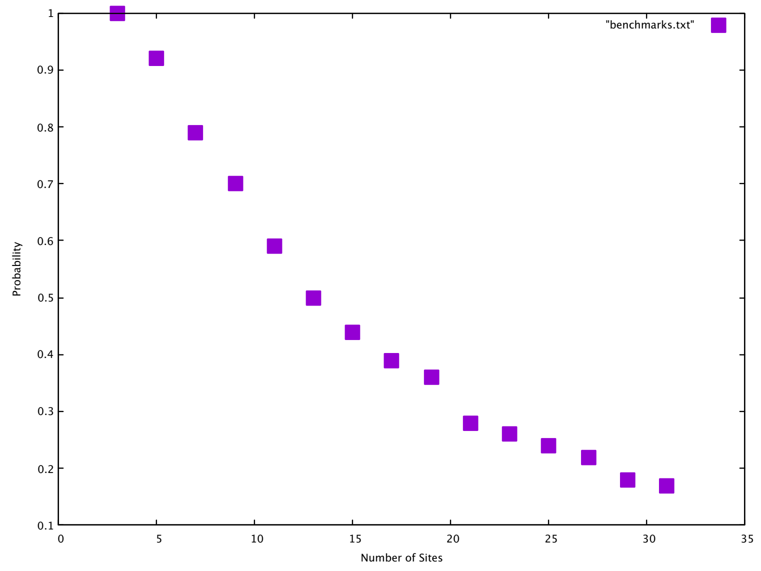


FIG. 1. Time Independent Benchmarks up to N=31

(e.g. 41,47) were calculated.

It is however clear that the grid-evaluation approach does not produce unique values of (T, β) corrisponding to the absolute probability maximum. On the other hand is able to give a qualitative idea on the probability distribution for various (T, β) combinations. We then present the found data with a **heatmap plot** such the one below, which makes quite easy to visualize the result.
figure: heatmap plot example

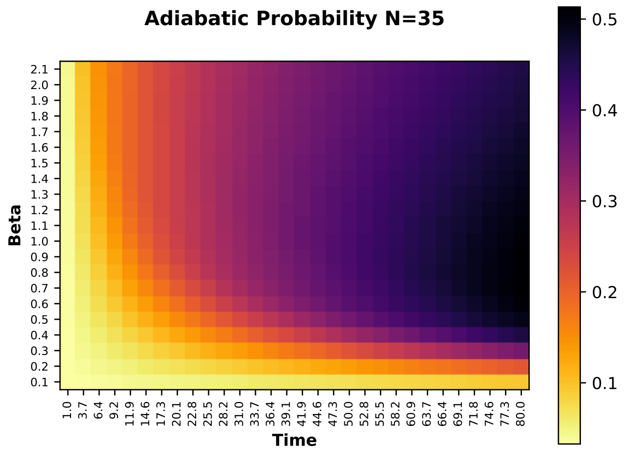


FIG. 2. Time Dependent heatmap example, N=35

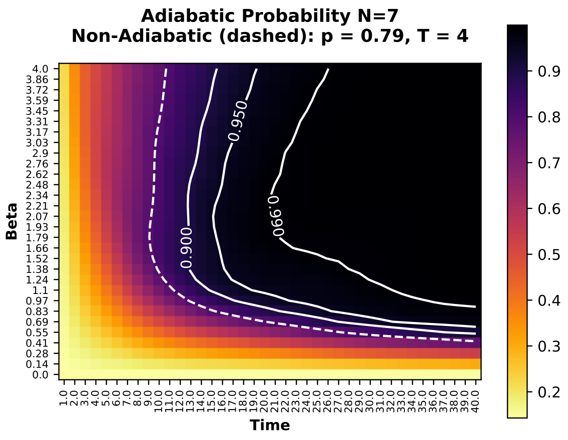
The task now is two develop a method to compare the time-independent and time-dependent algorithms. We first naively look at a qualitative comparison that immediately shows that the time-dependent algorithm is able to find higher probabilities but at the expense of worst time scaling; secondly we introduce a quantity to compare the two methods numerically.

3. Qualitative Comparison

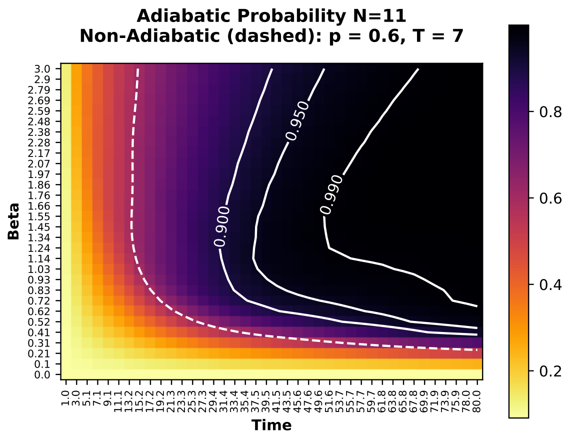
In order to qualitatively compare the two approaches to quantum search we will consider a heatmap plot where we superimpose probability **contour lines** corrisponding to the following probability (and with the following format):

- (solid) p = 0.99, p = 0.95, p = 0.90 for the time-dependent results
- (dashed) the probability found in the corrisponding time-independent results
- (dashed) double the probability (as long as it doesn't exceed unity) found for the time-independent result

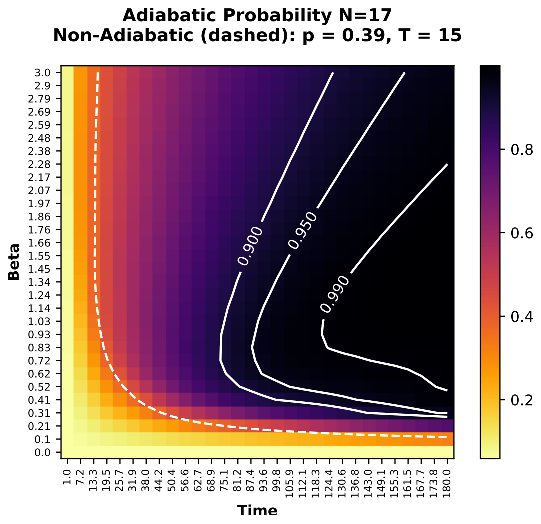
These, with the corrisponding time T at which the time-independent probability is found (given by a vertical line), should give a qualitative method to compare the two approaches, namely looking at the intersection between the first dashed line and the vertical time-line.
figure: heatmap plots with contour lines and vertical time-line. for N=7,11,17,21,27,31



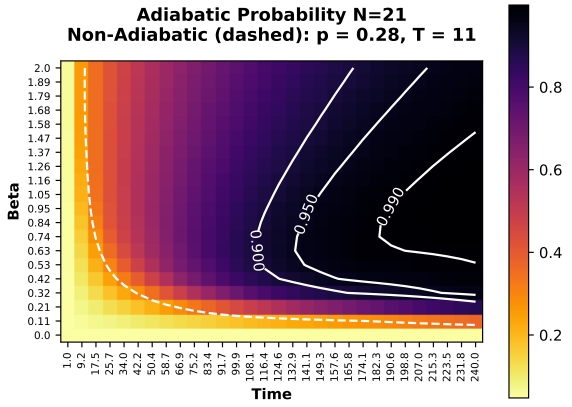
(a) N = 7



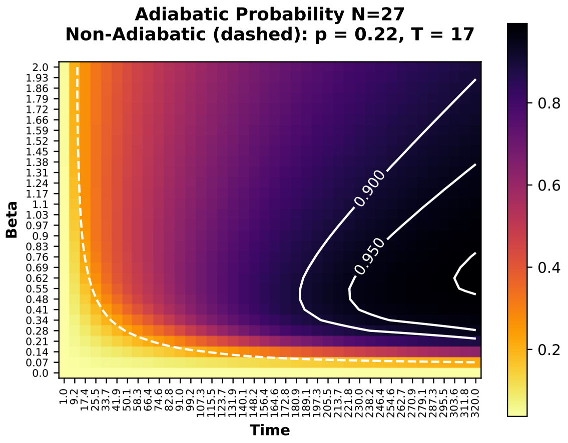
(b) N = 11



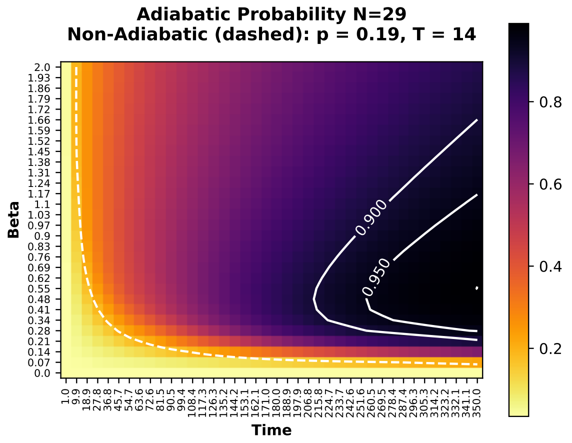
(a) N = 17



(b) N = 21



(a) N = 27



(b) N = 29

4. Numerical Comparison

In order to numerically compare the two approaches we define a quantity δ in the following way:

$$\delta = \min\left(\frac{T}{p}\right) \tag{17}$$

The quantity T/p represents the total time necessary to get to unitary probability. Finding the minimum of that quantity corresponds to finding the best combination of T and p that gets to unity with the minimum T.

What we do is thus for each vertical column of the grid, i.e. constant T and variable β, computing the maximum probability and finding the quantity T/p. We than find the minimum of all the T/p calculated. In this way we **reduce a grid to a single value** to compare to the time-independent results.

The following plot compares the time-independent approach to the time-dependent one, using for the first the quantity T/p evaluated with the β-T *optimized* probability while the latter uses the quantity δ.
figure: plot comparing two methods using the quantity δ

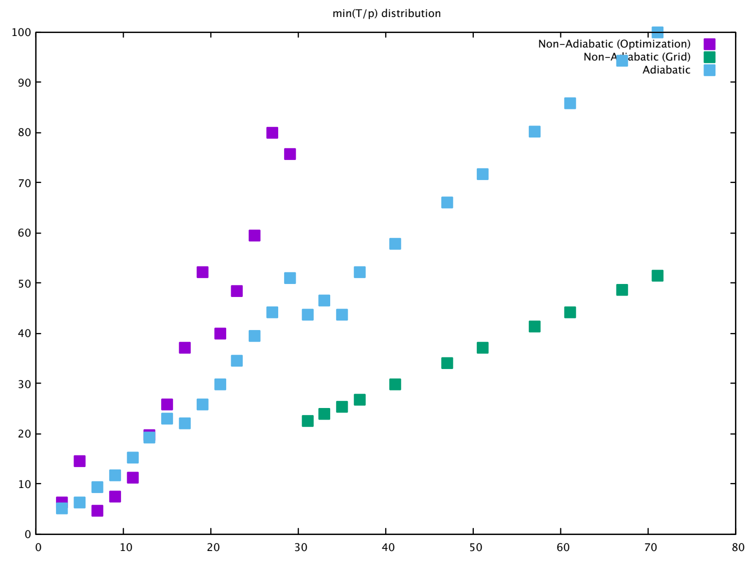


FIG. 3. delta values comparing methods

5. Different s(t) shapes

As we introduced in Section ? we're interested in determining how influencial is the shape of s(t) in the probability distribution. We then proceed as in Section ? using the step function s(t) previously discussed.
To compare we consider the quantity δ. Since we're interested in the trend for large N we ignore small N and only consider a few sampled N between 35-70.

As the plot shows however we do not see any speedup for functions that are not linear, on the contrary we do get poorer results. The time-independent implementation remains the best performing one, followed by linear, cubic root and square root. I wonder whether this approach was well-thought in the first place. As was already mentioned in Section ? we might want to consider what was mentioned by Cerf et al.

6. Localization

We can clearly see from the heatmap plots that we showed for the qualitative comparison that for large T the probabability gets bigger, and it's quite close to one for very large T, namely $T > 10N$. A further analysis is due, focusing on studying the probability distribution. Although it's clear from the adiabatic theorem and from the results

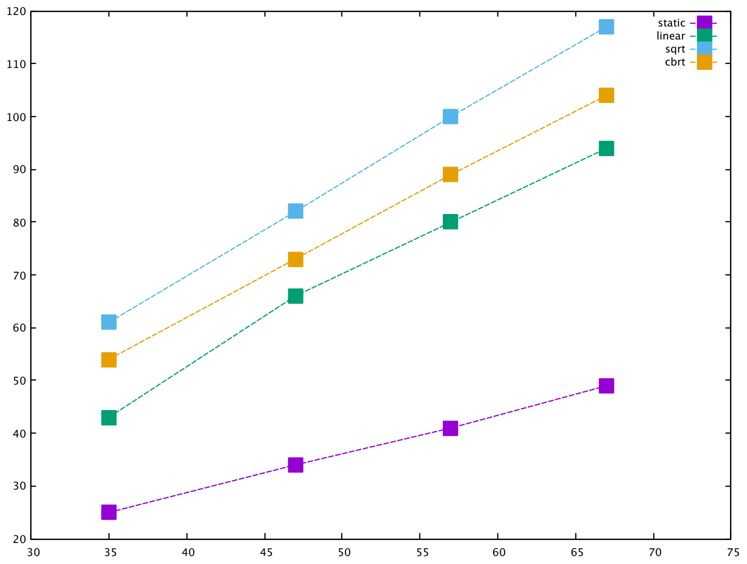


FIG. 4. Delta distrubution for different s(t). Computed for N = 35,47,57,67

that we currenly have that localization is a key feature of the time-dependent hamiltonian approach, further numerical simulations are necessary. In particular we might be interested in finding a trend (in terms of N) for few sampled probabilities, e.g. p = 0.5,0.75,0.8, 0.9, 0.95, 1

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, possibily (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 $\sqrt{N}$
figure: Probability distribution for fixed γ 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 γ-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 γ) in the static algorithm (std QWAlgorithm). Snippets of the parameters used follow.

```
from scipy.optimize import basinhopping
optimization = optimize.basinhopping()
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
e-3	**e-6**	**1.0198**	**-0.0198**	**19.5**
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