

Challenges of adiabatic quantum evaluation of NAND trees

Luís Tarrataca

Received: date / Accepted: date

Abstract The quantum adiabatic algorithm solves problems by evolving a known initial state towards the ground state of a Hamiltonian encoding the answer to a problem. Although continuous- and discrete-time quantum algorithms exist capable of evaluating tree graphs consisting of N vertexes in $O(\sqrt{N})$ time, a quadratic improvement over their classical counterparts, no quantum adiabatic procedure is known to exist. In this work we present a study of the main issues and challenges surrounding quantum adiabatic evaluation of NAND trees. We focus on a number of issues ranging from: (i) mapping mechanisms; (ii) spectrum analysis and remapping; (iii) numerical evaluation of spectrum gaps and (iv) algorithmic procedures. These concepts are then used to provide numerical evidence for the existence of a $\frac{N^2}{\log N^2}$ gap.

Keywords quantum adiabatic computation, tree evaluation, spectrum evaluation, numerical analysis

Mathematics Subject Classification (2000) 81P68 · 68Q05 · 68Q10 · 68Q12

This work was supported by CNPq CSF/BJT grant reference 301181/2014-4

L. Tarrataca
QCG/LNCC
Av. Getúlio Vargas, 333
Quitandinha, Petrópolis - RJ, 25651-075, Brazil
E-mail: luis.tarrataca@gmail.com

1 Introduction

A diverse number of relevant computational problems can be specified through NAND tree evaluation, *e.g.* from calculating the value of boolean expressions [21] to determining the outcome of games [12]. This type of problems consist of a binary tree with: (i) 2^d leaf nodes representing a binary string of length 2^d , respectively *i.e.* x_1, \dots, x_{2^d} ; (ii) where $d = \log_2 N$ is the depth of the tree; and (iii) for each depth level $0 \leq k \leq \log_2 N$ there exists a total of 2^k internal nodes representing a NAND logical gate over its respective children. The root evaluates to function $f_d(x_1, \dots, x_{2^d})$, where $f_d : \{0, 1\}^{2^d} \rightarrow \{0, 1\}$ is defined recursively as presented in Equation (1) [10].

$$f_d(x_1, \dots, x_{2^d}) = \begin{cases} \neg(f_{d-1}(x_1, \dots, x_{2^{d-1}}) \wedge f_{d-1}(x_{2^{d-1}+1}, \dots, x_{2^d})), & d > 0 \\ x & , d = 0 \end{cases} \quad (1)$$

The problem of evaluating graphs is closely related to computational search. Graph evaluation distinguishes itself by the need to determine the value of a node that is a function of other nodes. Classically evaluation requires $O(N)$ time [20] whilst randomized algorithms are capable running in $O(N^{0.753})$ [21]. Traditional approaches to quantum search focus on finding marked states [18]. This is optimal since there is a known lower bound of $\Omega(\sqrt{N})$ [6]. A continuous-time algorithm for evaluating NAND trees in $O(\sqrt{N})$ was provided in [16]. In [10] the authors showed that it was possible to convert the continuous-time algorithm to the discrete quantum oracle query using $N^{\frac{1}{2}+o(1)}$ queries. A quantum walk based method for evaluating every NAND formula in bounded-error $N^{\frac{1}{2}+o(1)}$ -time was presented in [11]. These outcomes were complemented in [3] which presented a nearly optimal discrete query quantum algorithm for evaluating arbitrary formulas of depth d using $O(\sqrt{N} \log^{d-1} N)$ queries. A discrete quantum walk version, specifically tailored to an AND-OR formulas of size N requiring $N^{\frac{1}{2}+o(1)}$ queries was presented in [4]. These methods stimulated the development of quantum algorithms for discrete-time minimax evaluation. A bounded-error quantum algorithm for evaluating minimax trees with $N^{\frac{1}{2}+o(1)}$ queries was introduced in [12].

Recently, advances in quantum devices that focus on solving optimization problems, has led to a renewed interest in the range of potential applications of adiabatic quantum computation. This paradigm of computation differs substantially from the methods that were previously discussed. Thus, it is natural to question how to perform NAND tree evaluation in an adiabatic context, more specifically: (i) how to develop an adequate approach?; (ii) is it possible to capitalize on existing methods?; (iii) what are the main advantages and disadvantages associated with each procedure?; (iv) what are the potential running times of potential methods?; amongst other issues.

1.1 Organization

The remainder of this work is organized as follows. Section 2 presents an overview of adiabatic quantum computation; Section 3 describes current methods to quantum NAND tree evaluation. Section 4 discusses how to develop an adiabatic mapping. Section 5 presents the main conclusions of this work and potential implications.

2 Adiabatic Computation

Adiabatic quantum computation was first introduced in [17]. This computational paradigm was later shown to be equivalent to standard model of quantum computation in [1]. The adiabatic procedure evolves a quantum system in a state $|\psi(t)\rangle$ according to the Schrödinger equation $i\hbar\frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle$. The Hamiltonian is a Hermitian matrix that can be perceived from a computational perspective as mapping a state to an energy value. The lowest value of the Hamiltonian's energy spectrum is referred to as its ground state. The adiabatic theorem [8] states that a quantum system with a time-dependent Hamiltonian that is initially in a certain energy level tends to stay at the same level, provided that the Hamiltonian is evolved slowly enough. $H(t)$ is a linear interpolation of Hamiltonians H_0 and H_P , as illustrated in Equation (2) [17]. H_0 is an initial Hamiltonian with a simple to prepare ground-state. H_P is a problem-specific Hamiltonian whose ground-state encodes a solution.

$$H(t) = (1 - \frac{t}{T})H_0 + \frac{t}{T}H_P \stackrel{s=\frac{t}{T}}{=} \tilde{H}(s) = (1 - s)H_0 + sH_P \quad (2)$$

If a parameter $s = \frac{t}{T}$ with $s \in [0, 1]$ is introduced then it is possible the right hand side of Equation (2). The introduction of such a parameter allows for the Schrödinger equation to be expressed in terms of s as described in Equation (3), where the fraction $\frac{dt}{ds}$ is usually described as a delay factor $\varrho(t)$ that controls the Hamiltonian's rate of change [13].

$$\frac{d}{ds}|\psi((s))\rangle = -i\frac{dt}{ds}\tilde{H}(s)|\psi(s)\rangle = -i\varrho(s)\tilde{H}(s)|\psi(s)\rangle \quad (3)$$

Let the instantaneous eigenstates and eigenvalues of $\tilde{H}(s)$ be defined as described in Equation (4) [17], where $|l; s\rangle$ denotes the instantaneous eigenstate associated with energy level l at time s and with $E_0(s) \leq E_1(s) \leq \dots \leq E_{N-1}(s)$ (index $l = 0$ represents the lowest energy level of the system, *i.e.* the ground state).

$$\tilde{H}(s)|l; s\rangle = E_l(s)|l; s\rangle \quad (4)$$

The adiabatic theorem states that if the gap $g(s) = E_1(s) - E_0(s)$ is greater than zero $\forall s \in [0, 1]$, then evolving the system for time s will produce a state $|\psi(s)\rangle$ that is very close to the instantaneous ground state of $\tilde{H}(s)$, provided that $\lim_{T \rightarrow +\infty} |\langle l = 0; s = 1 | \psi(T) \rangle| = 1$.¹ However, practical computation is not compatible with infinite running times. As a result, it is crucial to determine an appropriate lower bound for the evolution time allowing for adiabaticity. The adiabatic theorem states that choosing a delay schedule $\varrho(s)$ with the form $\varrho(s) \gg \frac{\|\frac{d}{ds}\tilde{H}(s)\|_2}{g(s)^2}$ is sufficiently slow for the evolution to remain adiabatic. Unfortunately, $g(s)$ is often a difficult task for most Hamiltonians. In those cases the minimum gap $g_{min} = \min_{0 \leq s \leq 1} (E_1(s) - E_0(s))$ is used instead. As noted in [13,17] the numerator is proportional to N , as a result, it is often customary to consider a time $T \gg \frac{1}{g(s)^2}$ or $T \gg \frac{1}{g_{min}^2}$.

3 Current approaches to quantum NAND tree evaluation

Given the adiabatic theorem, a natural question arises: how can the value of a NAND tree be assessed. One possibility would be to try to devise a method based on the corresponding adjacency matrix H . In the case of undirected graphs, such as trees, this type of matrices are symmetric which also implies that the operator is hermitian. This method would require a formal analysis of the operator in order to try to determine if any kind of useful information could be encoded into the eigenstates. Specifically, we would need to consider how the spectrum of operator $\tilde{H}(s)$ with $H_P = H$ would be affected depending on whether the tree evaluated to true or false.

As illustrated by the current literature ([16,4]), there is no trivial method to do this. Furthermore, it is often a complex task to determine which eigenstates convey useful information. Consequently, it is preferable to try to capitalise on existing methods for quantum NAND tree evaluation and assess their respective potentials for adiabatic mappings. There are essentially two methods: one based on discrete-time [4] and another relying on continuous-time [16]. Section 3.1 focuses on the former whilst Section 3.2 discusses the latter.

3.1 Discrete-time NAND Trees

The discrete procedure is based on a random walk on an extended graph containing a binary NAND tree and two additional nodes besides the root r , respectively, r' and r'' . Leaf vertices evaluating to 1 act as probability sinks and the transition probabilities at r' are biased towards the rest of the tree.

¹ Although this formulation requires a strictly greater than zero gap, subsequent version of the theorem were developed without the traditional gap condition [5]. The remainder of this work focuses on the well-established original method.

Controlling these transitions is operator $H|v\rangle = h_{pv}|p\rangle + \sum_c h_{vc}|c\rangle$ representing a symmetric, weighted adjacency matrix of the graph of the tree and the two additional nodes [4], where p is the parent of node v and the c represent a child of v . The weights for most part of the graph depend on the structure of the tree and have value $h_{pv} = (s_v/s_p)^{1/4}$, where s_k represents the number of inputs of the sub formula under node k . The authors are able to demonstrate that when the tree evaluates to zero, then there exists a zero-energy eigenvector $|a\rangle$ with an overlap on r'' , *i.e.* $\langle r''|a\rangle \geq 1/\sqrt{2}$. If the tree evaluates to 1 then any zero-energy eigenstates have no overlap with r'' . However, there may still exist other zero-energy eigenstates supported on other nodes. After \sqrt{N} steps of the quantum walk it is possible to perform quantum phase estimation to determine the value of the tree. The tree has value zero *iff* if the measured phase is 0 or π .

Although this method is able to encode information regarding the value of the tree on zero-energy eigenstates, at any specific point in time there may be several of such states, with not all of them conveying useful information regarding the value of the tree. Accordingly, basing any adiabatic method on such a procedure would be inherently problematic. This means that it is better to consider other alternatives to such a procedure.

3.2 Continuous-time NAND Trees

The continuous-time algorithm of [16] employs a wave-packet alongside a graph pathway with r labelling sites. The inverted NAND tree is connected to the pathway at index $r = 0$. An additional line of nodes is attached alongside the leafs. If an input has value one, a connection is established between the corresponding leaf and the additional node. Otherwise, the leaf remains unconnected. The algorithm constructs a plane wave packet $|\psi(0)\rangle$ corresponding to a right-moving packet that starts on the left of the pathway, with energy $E(\theta) = -2\cos\theta$, for $\theta > 0$, and that propagates with speed two. By employing quantum scatter theory the authors were able to characterize the energy eigenstates, $|E\rangle$, of $|\psi(0)\rangle$. This led to the conclusion that the packet continues moving to the right if the transmission coefficient of the scatter process, respectively $T(E)$, is approximately zero.

Further examination revealed that the transmission coefficient is a function of the amplitude ratio between the nodes $r = 0$ and the root of the tree as described in Equation (5).

$$T(E) = \frac{2i \sin \theta}{2i \sin \theta + y(E)}, \quad y(E) = \frac{\langle \text{root} | E \rangle}{\langle r = 0 | E \rangle} \quad (5)$$

The authors were also able to demonstrate that for states with energy close to zero, respectively $E \rightarrow 0^+$, $y(E)$ assumes one of two possible values, either

$y(E) = 0$ or $y(E) = -\infty$. At the core of such an analysis of the tree is a subgraph where each node is connected to two nodes above and one below, with the exception of the top and the bottom of the overall graph. This subgraph is illustrated in Figure 1.

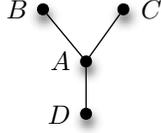


Fig. 1: The core subgraph of the binary NAND tree where every node is connected to two nodes above and to one below, with the exemption of the top and bottom layers (adapted from [16]).

Let a , b , c and d be the amplitudes of, respectively, the nodes $|A\rangle$, $|B\rangle$, $|C\rangle$ and $|D\rangle$ at an eigenvector $|E\rangle$. Furthermore, let H be the negative adjacency matrix [19]. Applying H at node A results in the value $Ea = -b - c - d$. From these equations it is possible to derive Equation (6) which represents a recursion based on the ratios $Y = a/d$, $Y' = b/a$ and $Y'' = c/a$.

$$\frac{a}{d} = -\frac{1}{E + \frac{b}{a} + \frac{c}{a}} \quad (6)$$

The connections between input variables x_1 and x_2 alongside their respective implications on the transmission factor are illustrated in Table 1. Notice that the transmission coefficient of Equation (5) assumes a behaviour that is identical to the NAND gate

x_1	x_2	Y'	Y''	$\lim_{E \rightarrow 0^+} Y$	$T(E)$	$\overline{x_1 \wedge x_2}$
0	0	$-\frac{1}{E}$	$-\frac{1}{E}$	0	1	1
0	1	$-\frac{1}{E}$	$\frac{1}{1-E^2}$	0	1	1
1	0	$\frac{E}{1-E^2}$	$-\frac{1}{E}$	0	1	1
1	1	$\frac{E}{1-E^2}$	$\frac{E}{1-E^2}$	$-\infty$	0	0

Table 1: The different connections between binary inputs, amplitude ratio $y(E)$, transmission coefficient and the equivalence to the NAND gate.

This method has better potential for an adiabatic mapping since zero-energy states, or at least those “close enough” to them, contain pertinent information regarding the value of the tree. This contrasts with the discrete method where zero-energy states also contain important information, but there could be a large number of them at the end of the procedure.

4 Adiabatic mapping

The continuous-time method offers the best potential for a successful mapping. However, a significant number of issues need to be addressed. The first of these issues is to consider what happens with the spectrum of the adjacency matrix representing the extended graph. Section 4.1 discusses these matters and also presents arguments for a spectrum remapping, which is presented in Section 4.2; Section 4.3 discusses why such a mapping is still in agreement with the NAND gate; Section 4.4 discusses the case where $\tilde{H}(s) \approx H_P^2$; Section 4.5 presents arguments for not performing the complete time evolution. Section 4.6 discusses the numerical analysis of the gap. The overall details of the adiabatic mapping procedure are presented in Section 4.7.

4.1 Tree Spectrum

The Hamiltonian NAND tree quantum algorithm relies on computational and scattering properties. However, it is not clear how one should go about mimicking the quantum scattering that occurs in the original method in an adiabatic fashion. With the adiabatic procedure we have the possibility to either minimise or maximise an energy function, depending on how the problem is formulated. This means that it is important to be able to encode in these states practical information. As a result, any possible adiabatic method for NAND tree evaluation needs to consider the spectrum of the associated Hamiltonian and how it is influenced by the results of the evaluation. From Section 3.2 we know that the extended adjacency matrix allows for the amplitudes to recurse down the tree in accordance with NAND logic. Accordingly, rather than trying to mimic the exact behaviour of the continuous-time method we will focus instead on what happens with energy levels close to zero.

Doing this requires a careful examination of the spectrum of the tree. Trees are bipartite graphs which have the interesting property that its spectrum is symmetric around zero. More specifically, suppose that the bipartite graph τ has an eigenvalue λ of multiplicity $m(\lambda)$. Then $-\lambda$ is also an eigenvalue of τ , and $m(-\lambda) = m(\lambda)$ [7]. The addition of the auxiliary node $r = 0$ connected to the root does not alter these properties since the graph remains bipartite. Let S denote the spectrum of the tree. An adequate representation of S is $S = \{-\lambda_k, -\lambda_{k-1}, \dots, \lambda_0, \dots, \lambda_{k-1}, \lambda_k\}$, where k is some constant and λ_0 represents the middle of the spectrum. The existence of λ_0 depends on whether or not the tree has zero-energy eigenstates.

From the perspective of adiabatic quantum computation this raises an important issue. Namely, the adiabatic procedure will return a λ -energy state that is different than zero, either $\max_{\lambda} S$ or $\min_{\lambda} S$, depending on how the problem is formulated. However, in the context of NAND tree evaluation we are interested in examining those states with energy zero or close to zero. This involves two

important aspects: (i) the existence of zero-energy eigenstates and (ii) how to obtain those zero-energy eigenstates at the end of the adiabatic procedure. We will address the former immediately and leave the latter to Section 4.2. The existence of zero-energy eigenstates can be demonstrated given the extended adjacency matrix of the tree plus node $r = 0$.

Proof: In order to prove that there will exist a zero-energy state please consider an Hilbert space, as described in [16], of dimension $\sum_{k=0}^d b^k + b^d + 1$ capable of accommodating the original tree, a base node $r = 0$ and an auxiliary line of input nodes. Figure 2 exemplifies: (i) the extended graph of a NAND tree associated with an additional line of nodes representing the input set $\{0, 1, 0, 1\}$ and evaluating to False; and (ii) the corresponding 12×12 adjacency matrix whose magnitude is $\left\{ \sqrt{\frac{1}{2}(5 + \sqrt{21})}, \frac{1}{2}(1 + \sqrt{5}), 1, \frac{1}{2}(\sqrt{5} - 1), \sqrt{\frac{1}{2}(5 - \sqrt{21})}, 0 \right\}$.

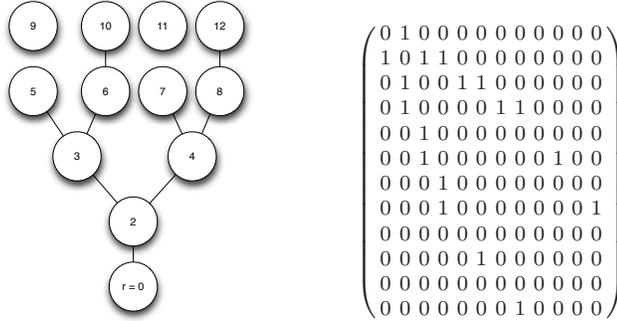


Fig. 2: A tree example alongside the respective adjacency matrix

Let H represent the adjacency matrix of the subgraph of Figure 1. This subgraph is important since every tree ultimately funnels the recursion amplitudes through a similar structure. In addition, it also serves to demonstrate the existence of zero-energy eigenstates. Let $H_C = |A\rangle(\langle B| + \langle C| + \langle D|)$, then it is possible to construct an eigenstate in such a way as to cancel amplitudes resulting in a zero energy state, *e.g.*:

- $H_C(|B\rangle - |C\rangle) = (|A\rangle\langle B| + |A\rangle\langle C| + |A\rangle\langle D|)(|B\rangle - |C\rangle) = |A\rangle - |A\rangle = 0$;
- $H_C(|B\rangle - |D\rangle) = (|A\rangle\langle B| + |A\rangle\langle C| + |A\rangle\langle D|)(|B\rangle - |D\rangle) = |A\rangle - |A\rangle = 0$;
- $H_C(|C\rangle - |D\rangle) = (|A\rangle\langle B| + |A\rangle\langle C| + |A\rangle\langle D|)(|C\rangle - |D\rangle) = |A\rangle - |A\rangle = 0$;

Furthermore, the same type of analysis can be performed when dealing with input states. Let $S := \{x_k = 0, \forall k \in [0, 2^n - 1]\}$ denote the set of input states x_k with zero value. Recall that the input nodes are connected to the corresponding parents if they have an input value of 1, otherwise no connection is made in the corresponding indexes of the adjacency matrix. As a result, the

corresponding adjacency matrix will have $|S|$ lines and rows of zeros which, in the presence of the appropriate eigenstates, will guarantee the existence of zero-energy states. This can be perceived by considering the adjacency matrix H_G of a graph G with the form described in Equation (7), where E represents the set of edges of the tree.

$$H_G|x_k\rangle = \left(\sum_{(u,v) \in E} |u\rangle\langle v| \right) |x_k\rangle \quad (7)$$

Assuming $|S| > 0$, then for $\forall x_k \in S$ the inner product $\langle v|x_k\rangle$ present in each individual term of the summation will be zero. This is a consequence of $\forall_{(u,v) \in E} v \neq x_k$, thus resulting in zero-energy states. Evidently, this condition does not hold when all inputs have value one. Determining whether or not all inputs are set would require $O(N)$ time. Theoretically, this poses a problem. There are two approaches that can be taken regarding this issue. First, most graph-based quantum algorithms work on the assumption that the adjacency matrix encoding the edges of the graph is somehow provided. Building such a structure has an inherent computational cost of $O(N)$. As a result, the methods developed so far have focused on evaluating the corresponding adjacency matrix after it has somehow been obtained. Verifying whether all inputs are one could be factored in whilst constructing the corresponding adjacency matrix, thus having negligible implications in overall time execution. Alternatively, the costs associated with such a verification could be mitigated by assuming that the algorithm would output a correct answer with high probability for large N .

4.2 Spectrum Remapping

Given the wide spectrum of H_P and the fact that the relevant information is contained in $E = 0$ it is natural to question how exactly is it possible to devise an adiabatic method that focuses specifically on those states. The main issue with the adiabatic procedure is how exactly to “zoom in” on certain states that are known to contain important information but do not necessarily correspond to either minimums or maximums. Unfortunately, there is no trivial way to develop such a mechanism since it requires a spectrum remapping.

However, it is possible to focus precisely on those cases where we are interested in obtaining the zero-energy states. Ideally, it would be desirable to formulate a method where the negative eigenvalues λ would be mapped to the positive spectrum by some η , *i.e.* $|\eta\lambda|, \forall \lambda < 0$. The same procedure could also be applied to positive values, *i.e.* $|\zeta\lambda|, \forall \lambda > 0$. The outcome of such a procedure would be a remapping of the spectrum. This would result in the set of non-negative real values, including zero, *i.e.* \mathbb{R}_0^+ assuming a real-valued spectrum.

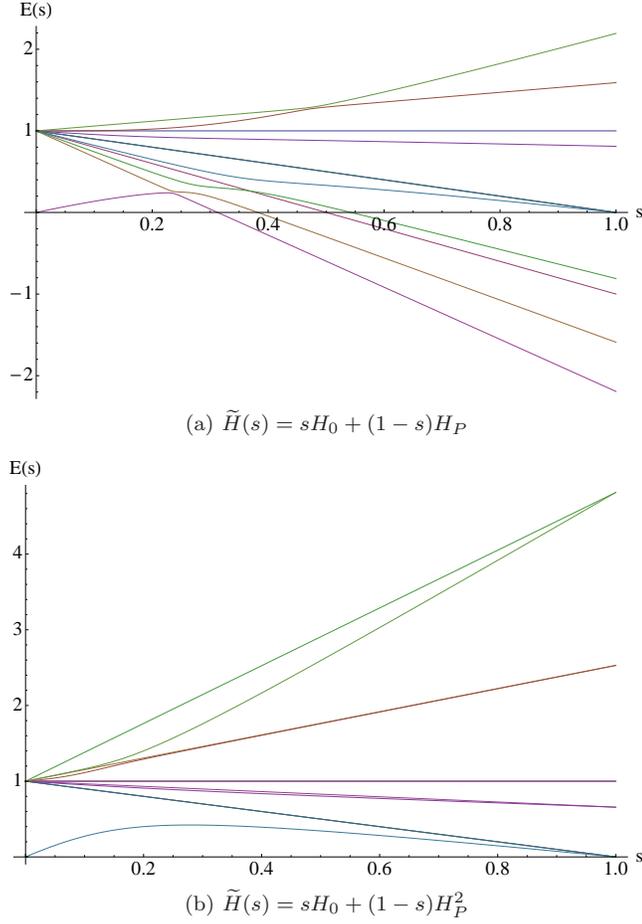


Fig. 3: Spectrum differences for a randomly generated NAND tree ($N = 4$).

The minimum energy of the spectrum would thus be zero, which is precisely the set of eigenstates we are interested in analysing. This can be performed if instead of applying operator $H_P = H$ we apply $H_P^2 = H^2$ which will perform the aforementioned remapping but with $\eta = \zeta = \lambda$. More specifically, given an operator H with eigenvector $|\psi\rangle$ and associated eigenvalue λ , then $|\psi\rangle$ is also an eigenvector of H^2 with corresponding eigenvalue λ^2 . Furthermore, because H is Hermitian H^2 will also have this property. The results of using H_P and H_P^2 in the interpolation \tilde{H} for an adjacency matrix representing a binary tree of depth 2 are presented, respectively, in Figure 3a and Figure 3b. For the full evolution time, $s = 1$, the spectrum of the interpolation for Figure 3a is symmetric since at this time $\tilde{H}(s) = H_P = H$.

Let H_0 have the form $H_0 = I - |\psi\rangle\langle\psi|$. It is also interesting to point out that in the case of Figure 3a there are several energy levels that cross, which would

immediately pose a problem in the traditional formulation of the adiabatic theorem. However, in this specific case $|E_1(s) - E_0(s)| > 0$, which is not always the case. Furthermore, at the end of the evolution procedure illustrated in Figure 3a the spectrum exhibits the characteristic symmetry of bipartite graphs. Although it is not possible to determine a general expression for $E_0(s)$, simulations for search spaces up to 32 input bits appear to indicate that in most cases $E_1(s) = 1 - s$. In addition, it should also be stated that, in a general manner, the spectrum for H_P^2 appears to follow curves similar to the ones illustrated in Figure 3b.

One should also be careful to mention that in the case of Figure 3b the two lowest eigenvalue curves coincide with each other only at $s = 1$. This is reminiscent of the case studied in [2] where it was shown that it was possible to devise a spectrum where an exponentially small gap existed very close to $s = 1$. Such a gap would preclude the existence of an efficient adiabatic algorithm. This problem was solved in [14] by randomising the choice of H_0 .

4.3 NAND Evaluation for H_P^2

Although employing H_P^2 succeeds in remapping the spectrum to a set of non-negative values one still needs to consider if the amplitudes continue to recurse down the tree in accordance with the NAND gate. This can be performed by first demonstrating that the overall recursion is still maintained for H_P^2 when applied to node A from Figure 1 by studying how the different amplitudes behave in regard to the eigenvectors. Equation (6) illustrates this procedure, which follows the same methodology from [16].

$$\begin{aligned} \langle A|H_P^2|E\rangle &= \begin{cases} (-\langle B| - \langle C| - \langle D|)H_P|E\rangle = E(-b - c - d) \\ \langle A|H_P|E\rangle = E\langle A|H_P|E\rangle = E^2a \end{cases} \\ &\Leftrightarrow E(-b - c - d) = E^2a \\ &\Leftrightarrow \frac{a}{d} = \frac{-1}{E + \frac{b}{a} + \frac{c}{a}} \end{aligned} \quad (8)$$

Equation (8), describes the overlap existing between the eigenvector $|E\rangle$ and the state that results from applying operator H_P^2 to state A . There are two possible ways to describe such a transition, which enables us to characterize the recursion behaviour, which is the same of Equation (6). Once the overall recursion is obtained one merely needs to show that the overall values for Y' and Y'' described in Table 1 are still observed. This procedure can be performed by focusing on either node $|B\rangle$ or $|C\rangle$ and demonstrating demonstrating whether the input node is connected or not. Accordingly, lets consider what happens with node B in the absence of the corresponding edge to the

associated input node. This procedure is illustrated by Equation (9), matching the original behaviour of H_P .

$$\begin{aligned} \langle B|H_P^2|E\rangle &= \begin{cases} \langle B|H_P H_P|E\rangle = -|A\rangle H_P|E\rangle = -Ea \\ \langle B|H_P E|E\rangle = E|B\rangle H_P|E\rangle = E^2b \end{cases} \\ &\Leftrightarrow E^2b = -Ea \\ &\Leftrightarrow \frac{b}{a} = -\frac{1}{E} \end{aligned} \quad (9)$$

The same rational can be applied to verify the behaviour when an input node I_1 is connected to node B yielding Equation (10) and Equation (11).

$$\begin{aligned} \langle B|H_P^2|E\rangle &= \begin{cases} \langle B|H_P H_P|E\rangle = (-|I_1\rangle - |A\rangle)H_P|E\rangle = (-i_1 - a)E \\ \langle B|H_P E|E\rangle = E|B\rangle H_P|E\rangle = E^2b \end{cases} \\ &\Leftrightarrow E^2b = (-i_1 - a)E \\ &\Leftrightarrow \frac{b}{a} = -\frac{1}{E + \frac{i_1}{b}} \end{aligned} \quad (10)$$

$$\begin{aligned} \langle I_1|H_P^2|E\rangle &= \begin{cases} \langle I_1|H_P H_P|E\rangle = -|B\rangle H_P|E\rangle = -Eb \\ \langle I_1|H_P E|E\rangle = E|I_1\rangle H_P|E\rangle = E^2i_1 \end{cases} \\ &\Leftrightarrow -Eb = E^2i_1 \\ &\Leftrightarrow \frac{i_1}{b} = -\frac{1}{E} \end{aligned} \quad (11)$$

By combining both of these expression it is possible to derive the $\frac{E}{1-E^2}$ fraction associated with a connection to an input node. Thus the overall behaviour for H_P^2 is the same as for H_P and the amplitudes continue to recurse down the tree in accordance with the NAND gate. As a result, the ratio $Y = a/d$, whose values will be present in the eigenstate, will continue to encode information about the value of the tree in H_P^2 .

4.4 The case for $\tilde{H}(s) \approx H_P^2$

Fortunately, we do not need to perform full adiabatic evolution since [16] proved that the system starts exhibiting its properties when $E \in]0, \frac{1}{256N}[$. Accordingly, this means that it is possible to stop the adiabatic evolution as soon as the ground state of $\tilde{H}(s)$, respectively $E_0(s)$, reaches the required threshold. Notice that there are two possible cases when $E_0(s) \leq T$, namely, at the beginning and end of the evolution. Intuitively, the former will exhibit

mostly behaviour associated with hamiltonian H_0 , whilst the latter, will result in an interpolation that is closer to H_P^2 (for $s > 1/2$). This intuition can be demonstrated from a mathematical perspective by proving that the overall interpolation remains close enough to the behaviour of the NAND gate when s approaches value 1, *i.e.* $s \rightarrow 1^-$. Initially, the interpolation $\tilde{H}(s)$ starts out with the n -dimension Hamiltonian $H_0 = I - |\psi_0\rangle\langle\psi_0|$, which can be expressed as presented in Equation (12), where $\delta = 1 - \frac{1}{\sqrt{N}}$ and $\sigma = \frac{1}{\sqrt{N}}$.

$$H_0 = \begin{pmatrix} \delta & \sigma & \dots & \sigma \\ \sigma & \delta & \dots & \sigma \\ \vdots & \vdots & \ddots & \vdots \\ \sigma & \sigma & \dots & \delta \end{pmatrix} \quad (12)$$

Please consider the core graph element from Figure 1 but with an additional line of input nodes i_1 and i_2 . At any given point in time s the interpolation can be perceived as adding a certain amount of noise to H_P . Since our target Hamiltonian is an adjacency matrix, the noise introduced can be perceived as altering the connections of the tree by introducing certain edges that were not initially present. As a result, the overall behaviour of $\tilde{H}(s)$ when applied to eigenstates can be described as illustrated in Equation (13), where: (i) the lower-case letters represent the amplitudes at the respective nodes; and (ii) $E(s)$ represents the energy at time s of eigenstate $|E\rangle$.

$$\begin{aligned} \lim_{s \rightarrow 1} \langle A | \tilde{H}(s) | E \rangle &= \begin{cases} \lim_{s \rightarrow 1} [(1-s)\sigma - s](b+c+d) + (1-s)(\delta a + \sigma(i_1 + i_2)) \\ \lim_{s \rightarrow 1} E(s)a \end{cases} \\ &= \begin{cases} -(b+c+d) \\ E(1)a \end{cases} \\ &\Leftrightarrow \frac{a}{d} = -\frac{1}{E(1) + \frac{b}{a} + \frac{c}{a}} \end{aligned} \quad (13)$$

In the specific case of our adiabatic evolution procedure where $H_P^2 = H^2$ then $E(1) = E_0(1) = 0^+$, *i.e.* given that the system will remain in the ground state, the energy $E(s)$ will be the energy $E_0(1)$ which will be close to zero. Thus, Equation (13) exhibits the same behaviour as Equation (6), despite using an interpolation Hamiltonian. Accordingly for $s \rightarrow 1$ the amplitudes continue to recurse down the tree in accordance with NAND logic.

4.5 The case for $s \approx 1$

The adjacency matrix H starts exhibiting its NAND properties when $E \in]0, \frac{1}{16\sqrt{N}}[$. As a result we want to try to determine what is the gap when

$E_0(s)$ reaches $\frac{1}{256N}$ for $s \approx 1$. Because it is not possible to determine an overall expression for $g(s)$ the only viable route is to perform some type of numerical analysis. In this case we opted to examine the average gap. The overall approximate results for the minimum gap g_{min} obtained when $E_0(s)$ reaches $\frac{1}{256N}$ for $s \approx 1$ are presented in Table 2. Also included in the table are: (i) the values for the respective minimum gap g'_{min} that is observed in the standard adiabatic search procedure; and (ii) the ratio between minimum gaps for our procedure and the general search mechanism in difference.

	$N = 2$	$N = 4$	$N = 8$	$N = 16$
g_{min}	0.000586502	0.000259816	0.000138516	0.0000486609
g'_{min}	0.707107	0.5	0.000138516	0.25
$\frac{g'_{min}}{g_{min}}$	1205.63	1924.44	2552.43	5137.59

Table 2: The minimum gaps when considering several search space dimensions, respectively $N = \{2, 4, 8, 16\}$ for: (i) H_P when $E_0(s)$ reaches $\frac{1}{256N}$ for $s \approx 1$ respectively g_{min} ; and (ii) the standard adiabatic search procedure, respectively g'_{min} . Also included is the ratio between both minimum gaps in order to demonstrate the growth between instances.

The first thing that draws immediate attention is the fact that the minimum gaps are very small, which was to be expected when one considers the original curve from Figure 3b. In the original quantum adiabatic search procedure the minimum gap was $N^{-\frac{1}{2}}$. This contrasts sharply with the values obtained for g_{min} which are significantly smaller and appear to be getting smaller with each instance of the dimensions considered. Given the reduced number of data points attempting to perform a fit will not produce a convincing result. Therefore, it is important to investigate other dimensions in order to determine if anything can be said about the impact of gap on performance.

4.6 Analysing the gap for H_P^2

Analysing the gap is crucial in order to determine an appropriate running time. Unfortunately, performing simulations is enough to demonstrate that there is no clear general expression for the gap condition. This can be exemplified by presenting the minimum gaps for the interpolations consisting of the bits $\{0, 0\}$ and $\{1, 1\}$ which are described in Table 3.

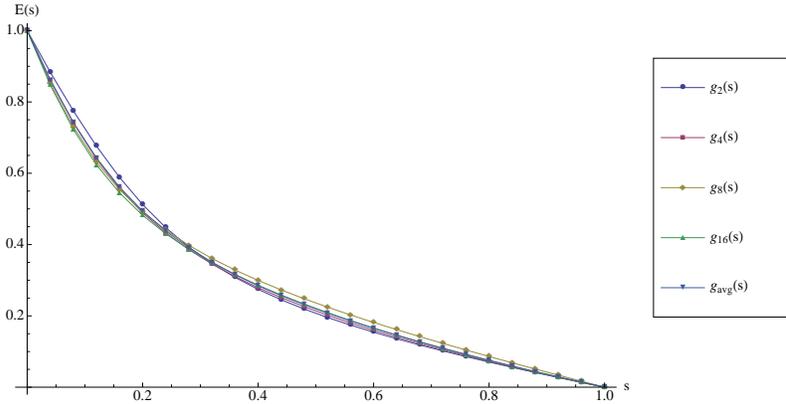
Lengthier bit sequences, and the associated increase in the number of dimensions, result in more complex expressions. This contrasts with the local adiabatic quantum search algorithm where the gap expression could be determined

x_1	x_2	$g(s)$
0	0	$\frac{1}{2} \left(\sqrt{12s^2 - 4s + 1} - 2s - 1 \right) - s + 1$
1	1	$\frac{1}{6} \left(\sqrt{3} \sqrt{55s^2 - 22s + 3} - 9s - 3 \right) - \sqrt{3}s + s + 1$

Table 3: Illustration of the gaps for two different bit sequences.

in a simple manner. As a result, the most that can be performed is a set of numerical simulations that try to give an overview of the minimum gap. However, given the exponential growth, in time and space, that is associated with Hilbert spaces of dimension 2^n , where n is the number of bits, we were only able to evaluate search spaces up to and including 16 bits. Furthermore, for $n = 16$ we opted to evaluate 50 random instances. This decision was made given the already considerable dimensions involved and their respective implications on computational time (similar to the one performed in [17] and [15]).

The data results from this procedure are illustrated in Figure 4 which plots the average gap condition $g_N(s)$, where N is the number of inputs alongside the overall average gap condition for all the input instances considered, respectively, $g_{avg}(s)$.

Fig. 4: Results for g_{avg} for all inputs consisting of 2, 4, 8 and 16 bits.

A number of observations can be immediately performed. First, the numerical plots exhibit relatively small variation with each increase in problem space size. Secondly, there exists a close resemblance between the data points and the exponential function with a negative power. This behaviour hints at the possibility of performing some-type of fitting. As is the case with any type of fitting procedure there may be multiple hypothesis which may not necessarily

correspond to the original function. One possibility is to attempt to perform a fit based on a polynomial gap decrease. However, finding a successful polynomial fit would have implications for the $P \stackrel{?}{=} NP$ question. As a consequence, an exponential decay represents a more realistic scenario for potential fits.

More specifically, given gap $g_k(s)$ it is possible to contemplate a fit based on function $a + b^{-s}$ where a and b are constants and s is the variable. The specific numerical results for such a fit are described in Table 4 (alongside fits for $a + b^{-\frac{s}{2}}$ and $a + b^{-\frac{s}{3}}$), whilst Figure 5 plots the fit of $g_{avg}(s)$ against the average data points for the dimensions considered. As a result, given the apparent good correlation between data points and the fit function, one possible hypothesis is for the gap function to exhibit this type of exponential decay.

	$a + b^{-s}$		$a + b^{-\frac{s}{2}}$		$a + b^{-\frac{s}{3}}$	
	a	b	a	b	a	b
$g_2(s)$	0.0118293	89.2008	0.0118293	7956.79	0.0140743	11313.5
$g_4(s)$	0.0165725	97.3933	0.0165725	9485.46	0.0212065	10241.4
$g_8(s)$	0.0168446	100.754	0.0168446	10151.3	0.0365949	4285.37
$g_{16}(s)$	0.0041261	141.183	0.00412613	19932.6	0.0310915	7252.96
$g_{avg}(s)$	0.0117078	103.268	0.0117078	10664.4	0.023929	8150.41

Table 4: Different fit values for functions $a + b^{-s}$, $a + b^{-\frac{s}{2}}$ and $a + b^{-\frac{s}{3}}$.

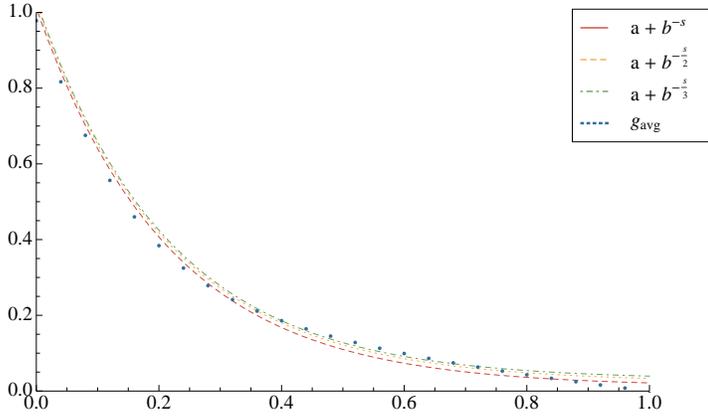


Fig. 5: The fit for $g_{avg}(s)$ for based on: (i) functions $a + b^{-s}$, $a + b^{-\frac{s}{2}}$ and $a + b^{-\frac{s}{3}}$, where a and b are constants and s is the variable; and (ii) the average data points for the dimensions considered.

From [9] we know that adapting the evolution rate of each infinitesimal time interval such that the evolution remains adiabatic allows one to obtain the

original quadratic speedup. Applying the same principle to our gap fits allows us to obtain a general expression for time t , respectively Equation (14).

$$t = \int_0^1 \frac{1}{g(s)^2} ds = \frac{\frac{1}{ab+1} + \log(ab+1) - \frac{1}{a+1} - \log(a+1)}{a^2 \log(b)} \quad (14)$$

Notice that the overall time t is a function of terms a and b . As a result, this expression can be further simplified since we know from the numerical data in Table 4 that the values for a are close to zero. Accordingly, by determining the appropriate limit for $a \rightarrow 0$ it is possible to find a simpler form for the overall time, as illustrated by Equation (15).

$$t \approx \lim_{a \rightarrow 0} \left(\frac{\frac{1}{ab+1} + \log(ab+1) - \frac{1}{a+1} - \log(a+1)}{a^2 \log(b)} \right) = \frac{b^2 - 1}{2 \log b} \Rightarrow O\left(\frac{b^2}{\log b}\right) \quad (15)$$

We opted to take into account only the most factors of Equation (15) since these are the ones that matter the most from a computational complexity perspective. The end result is a function of b , which, from Table 4, appears to be the most relevant variable in the exponential fit. Accordingly, determining an appropriate end result requires an adequate evaluation of how b is growing. Intuitively, for the instances considered, the data from Table 4 appears to be showing that b is growing quicker than linearly but less than the square power. As a result, in a worst-case scenario, if parameter b is modelled as a function of N then this would mean that b would be growing quadratically with N . Our choice for modelling b as a function of N is based on the fact that the dimension of the search space is the most relevant factor from a computational complexity perspective. Consequently, the final complexity of the adiabatic evolution is $O\left(\frac{N^4}{\log N^2}\right)$, which represents an overall performance penalty of $N^3/(2 \log N)$ when compared against the classical $O(N)$ time.

This penalty is mostly a consequence of the term N^4 , which itself is a consequence of the $g(s)^2$ factor employed during local adiabatic evolution. Clearly, the choice of exponent for the gap function has a crucial impact on the performance of the algorithm. As a result, if a fit of the type $a + b^{-s/2}$ is used instead (please refer to Table 4) then this would mean an overall complexity of $O\left(\frac{N^2}{\log N^2}\right)$. This value is still worse than the classical method, but represents an improvement over the initial bound.

The same type of reasoning can be used to assess the viability of any fit of the type $a + b^{-\frac{s}{k}}$ with $k \in \mathbb{Z}_1^+$. Figure 5 for the set illustrates the fits for the set of gaps $\{a + b^{-s/2}, a + b^{-s/3}, a + b^{-s/4}\}$, which would result, respectively, in complexities $O(N)$, $O\left(\frac{N^2}{\log N^2}\right)$, $O\left(\frac{N^{4/3}}{\log N^2}\right)$ and $O\left(\frac{N}{\log N^2}\right)$. Figure 6 illustrates how these complexities fare against each other. Notice that for $k = 3$ the

quantum fit starts exhibiting slightly better performance. However, as Table 4 demonstrates, for $k = 3$ it becomes difficult to interpret how the base of the exponent, respectively b , might be growing as a function of the dimension of the search space N . Increasing k beyond this point only serves to exacerbate the problem, with numerical simulations demonstrating the same type of difficulties with interpreting how b might be a function of N . Consequently, it may be conjectured that this behaviour could be indicative that the fit is no longer valid, which would imply worse than classical performance.

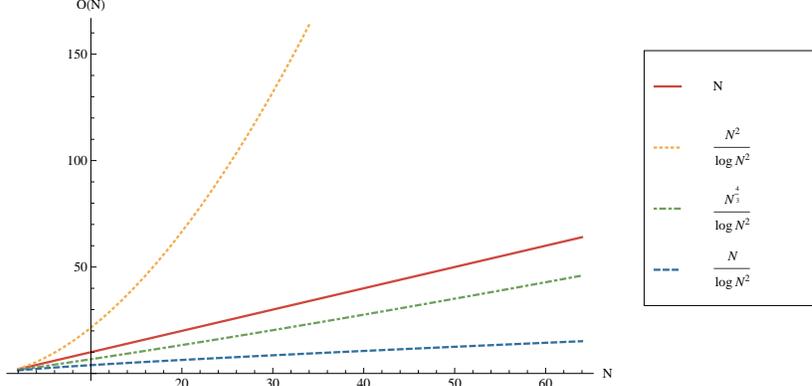


Fig. 6: Complexity functions plotted against $N \in [0, 64]$.

4.7 The overall procedure

Notice that although we know that H_P^2 behaves in accordance with the NAND gate for $E = 0^+$ we have yet to specify how could a potential adiabatic mapping be performed. In particular, once the final measurement of the quantum register is performed how does this translate to an evaluation value? In order to answer this question we would like to draw attention again to Table 1 and right hand side of Equation (5). In particular if the tree evaluates to:

- True then $\frac{\langle \text{root} | E \rangle}{\langle r=0 | E \rangle} = 0$, which means that $|\langle r = 0 | E \rangle|^2 \gg |\langle \text{root} | E \rangle|^2$. This means that the probability of measuring the node $r = 0$ is far greater than that of the root node.
- Alternatively, if the tree has False value then $\frac{\langle \text{root} | E \rangle}{\langle r=0 | E \rangle} = -\infty$. As a result this implies that $|\langle \text{root} | E \rangle|^2 \gg |\langle r = 0 | E \rangle|^2$.

The overall procedure for adiabatic tree evaluation is described in Algorithm 1 which receives as input a graph τ , consisting of: node $r = 0$, the original inverted tree and an additional line of nodes, alongside the values T and R

representing, respectively, the adiabatic evolution time and the number of times to repeat the experiment. The algorithm starts out by building the adjacency matrix employed in [16] in Line 2 and determining the corresponding dimension in Line 3. If everything is built correctly then $l = \sum_{k=0}^d b^k + b^d + 1$. The algorithm then proceeds to building the initial superposition state $|\psi_0\rangle$ (Line 4) followed by the H_0 operator, respectively Line 5. The overall interpolation is built in Line 6. The subspace operator is constructed in Line 7. In order to evaluate the result of the multiple experiments we also define and initialise to zero variables δ_{root} and $\delta_{r=0}$ in Lines 8-9.

Due to the probabilistic nature of quantum computation there is the need to repeat the experiment R times in order to assess with reliability the end result of the procedure. The path to obtain a good result for R can be found in [15]. However, given the set of possible values for Y (either $-\infty$ or 0) and the corresponding impact on the behaviour of the amplitudes for the root node and the $r = 0$ a fair assessment is that small values of R will produce a sufficiently good result. Each experiment iteration described in Line 10 covers a sample s of the experiment and evolves the system for time T in Line 11.

Once the evolution is concluded a measurement is performed on the subspace of the root node and node $r = 0$. This action is described in Line 12, where M refers to the measurement operator. Each sample s requires one to register the outcome of the experiment in the appropriate δ variable (Lines 13-16). The procedure concludes by performing a test in order to determine which node was obtained the most and outputting the appropriate result for the evaluation procedure (Lines 17-20).

Algorithm 1 Adiabatic Evaluation of the NAND tree

```

function  $f(\tau, T, R)$ 
   $H_P = \text{adjacencyMatrix}[\tau]$ 
   $l = \text{Length}[H]$ 
   $|\psi_0\rangle = \frac{1}{\sqrt{l}} \sum_{x=0}^{l-1} |x\rangle$ 
   $H_0 = I_{l \times l} - |\psi_0\rangle\langle\psi_0|$ 
   $H[t] = (1-t)H_0 + tH_P^2$ 
   $P = |\text{root}\rangle\langle\text{root}| + |r=0\rangle\langle r=0|$ 
   $\delta_{\text{root}} = 0$ 
   $\delta_{r=0} = 0$ 
  for  $s = 1$  to  $R$ 
     $|\psi_T\rangle = e^{-i \int_0^T H[t] dt} \cdot |\psi_0\rangle$ 
     $\xi = MP|\psi_T\rangle$ 
    if  $\xi == |\text{root}\rangle$ 
       $\delta_{\text{root}} = \delta_{\text{root}} + 1$ 
    else
       $\delta_{r=0} = \delta_{r=0} + 1$ 
  if  $\delta_{r=0} \gg \delta_{\text{root}}$ 
    return TRUE
  if  $\delta_{\text{root}} \gg \delta_{r=0}$ 
    return FALSE

```

5 Conclusions

In this work we presented a numerical study of the evaluation of NAND trees for up to 16 bit inputs through adiabatic quantum computation. Although the question of how to evaluate trees in an adiabatic manner efficiently, at least in comparison with existing quantum methods, remains an open one it is interesting to observe that there exist significant viability issues with developing mappings based on known algorithms. Of the issues encountered, perhaps the most important is the exponential gap decrease that is associated with the Hamiltonian composed of the adjacency matrix. This rapid decrease in gap is of such an order that even adapting the rate of evolution at each infinitesimal time interval is not enough to yield an improvement, or for that matter, even an equal running time. This assumes an exponential decrease whose basis can be expressed as a function of the dimension of the search space.

This seems to provide further evidence to the informal argument that computationally hard problems appear to have exponentially small gaps. Although, in this case, the gap derived from H_P^2 is even smaller than what would have been expected. In spite of the fact that, theoretically, it is possible to devise exponential fits that are able to outperform the classical variants the plausibility of these can be questioned. The exponential space required to represent the adjacency matrix also constitutes an inherent problem that is often times neglected to be referred. This sharply contrasts with Grover's algorithm that requires $O(\log_2 N)$ space.

Finally, we would like to emphasize that none of the aforementioned arguments rule out the theoretical possibility of devising an adiabatic quantum NAND tree evaluation algorithm more efficient than its classical counterparts. It only means that attempting to leverage existing methods does not appear to represent a viable strategy and may therefore require a completely different approach.

Acknowledgements

Luis Tarrataca's work was supported by CNPq CSF/BJT grant reference 301181/2014-4.

References

1. Aharonov, D., van Dam, W., Kempe, J., Landau, Z., Lloyd, S., Regev, O.: Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation. eprint arXiv:quant-ph/0405098 (2004)
2. Altshuler, B., Krovi, H., Roland, J.: Adiabatic quantum optimization fails for random instances of NP-complete problems. ArXiv e-prints (2009)

3. Ambainis, A.: A nearly optimal discrete query quantum algorithm for evaluating NAND formulas. ArXiv e-prints (2007)
4. Ambainis, A., Childs, A.M., Reichardt, B.W., Spalek, R., Zhang, S.: Any AND-OR Formula of Size N can be Evaluated in time $N^{\frac{1}{2}+o(1)}$ on a Quantum Computer. In: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, FOCS '07, pp. 363–372. IEEE Computer Society, Washington, DC, USA (2007). DOI 10.1109/FOCS.2007.10. URL <http://dx.doi.org/10.1109/FOCS.2007.10>
5. Avron, J.E., Elgart, A.: Adiabatic theorem without a gap condition. Communications in Mathematical Physics **203**(2), 445–463 (1999). DOI 10.1007/s002200050620. URL <http://dx.doi.org/10.1007/s002200050620>
6. Bennett, C.H., Bernstein, E., Brassard, G., Vazirani, U.: Strengths and Weaknesses of Quantum Computing. eprint arXiv:quant-ph/9701001 (1997)
7. Biggs, N.: Algebraic Graph Theory. Cambridge Mathematical Library. Cambridge University Press (1993). URL <http://books.google.com.br/books?id=6TasRmIF0xQC>
8. Born, M., Fock, V.: Beweis des adiabatensatzes. Zeitschrift für Physik **51**(3-4), 165–180 (1928). DOI 10.1007/BF01343193. URL <http://dx.doi.org/10.1007/BF01343193>
9. Cerf, N.J., Grover, L.K., Williams, C.P.: Nested quantum search and structured problems. Phys. Rev. A **61**(3), 032,303 (2000). DOI 10.1103/PhysRevA.61.032303
10. Childs, A.M., Cleve, R., Jordan, S.P., Yonge-Mallo, D.: Discrete-Query Quantum Algorithm for NAND Trees. Theory of Computing **5**(1), 119–123 (2009). DOI 10.4086/toc.2009.v005a005
11. Childs, A.M., Reichardt, B.W., Spalek, R., Zhang, S.: Every NAND formula of size N can be evaluated in time $N^{\frac{1}{2}+o(1)}$ on a quantum computer. eprint arXiv:quant-ph/0703015 (2007)
12. Cleve, R., Gavinsky, D., Yonge-Mallo, D.: Quantum algorithms for evaluating min-max trees. In: Y. Kawano, M. Mosca (eds.) Proceedings of Theory of Quantum Computation, Communication, and Cryptography (TQC 2008), vol. 5106, pp. 11–15. Springer Berlin / Heidelberg (2008)
13. van Dam, W., Mosca, M., Vazirani, U.: How powerful is adiabatic quantum computation? In: Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on, pp. 279–287 (2001). DOI 10.1109/SFCS.2001.959902
14. Farhi, E., Goldstone, J., Gosset, D., Gutmann, S., Meyer, H.B., Shor, P.: Quantum adiabatic algorithms, small gaps, and different paths (2009). URL <http://www.citebase.org/abstract?id=oai:arXiv.org:0909.4766>
15. Farhi, E., Goldstone, J., Gutmann, S.: A Numerical Study of the Performance of a Quantum Adiabatic Evolution Algorithm for Satisfiability. eprint arXiv:quant-ph/0007071 (2000)
16. Farhi, E., Goldstone, J., Gutmann, S.: A quantum algorithm for the hamiltonian nand tree. Theory of Computing **4**(1), 169–190 (2008). DOI 10.4086/toc.2008.v004a008
17. Farhi, E., Goldstone, J., Gutmann, S., Sipser, M.: Quantum Computation by Adiabatic Evolution. ArXiv Quantum Physics e-prints (2000)
18. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, pp. 212–219. ACM, New York, NY, USA (1996). DOI <http://doi.acm.org/10.1145/237814.237866>
19. Mochon, C.: Hamiltonian oracles. Phys. Rev. A **75**, 042,313 (2007). DOI 10.1103/PhysRevA.75.042313. URL <http://link.aps.org/doi/10.1103/PhysRevA.75.042313>
20. Russell, S., Norvig, P.: Artificial intelligence: a modern approach. Prentice Hall series in artificial intelligence. Prentice Hall (2010)
21. Saks, M., Wigderson, A.: Probabilistic boolean decision trees and the complexity of evaluating game trees. In: Proceedings of the 27th Annual Symposium on Foundations of Computer Science, SFCS '86, pp. 29–38. IEEE Computer Society, Washington, DC, USA (1986). DOI 10.1109/SFCS.1986.44