

Systematic analysis of majorization in quantum algorithms

R. Orús^{1,a}, J.I. Latorre¹, and M.A. Martín-Delgado²

¹ Dept. d'Estructura i Constituents de la Matèria, Univ. Barcelona, 08028 Barcelona, Spain

² Departamento de Física Teórica I, Universidad Complutense, 28040 Madrid, Spain

Received 28 August 2003 / Received in final form 11 December 2003

Published online 10 February 2004 – © EDP Sciences, Società Italiana di Fisica, Springer-Verlag 2004

Abstract. Motivated by the need to uncover some underlying mathematical structure of optimal quantum computation, we carry out a systematic analysis of a wide variety of quantum algorithms from the majorization theory point of view. We conclude that step-by-step majorization is found in the known instances of fast and efficient algorithms, namely in the quantum Fourier transform, in Grover's algorithm, in the hidden affine function problem, in searching by quantum adiabatic evolution and in deterministic quantum walks in continuous time solving a classically hard problem. On the other hand, the optimal quantum algorithm for parity determination, which does not provide any computational speed-up, does not show step-by-step majorization. Lack of both speed-up and step-by-step majorization is also a feature of the adiabatic quantum algorithm solving the 2-SAT “ring of agrees” problem. Furthermore, the quantum algorithm for the hidden affine function problem does not make use of any entanglement while it does obey majorization. All the above results give support to a step-by-step Majorization Principle necessary for optimal quantum computation.

PACS. 03.67.-a Quantum information – 03.67.Lx Quantum computation

1 Introduction

One of the main open problems in quantum computation theory is finding some mathematical structure underlying optimal quantum algorithms. There is a rather short list of ideas behind the design of fast algorithms. Grover's quantum searching algorithm [1] exploits calls to an oracle by enhancing a particular state via a rotation in the relevant Hilbert space associated to the problem. Shor's quantum factoring algorithm [2] exploits the periodicity of an initial quantum state using only a minimum of Hadamard gates at the core of the quantum Fourier transform. Based on more general quantum mechanical principles, the idea of using adiabatic evolution to carry quantum computation [3] has proven suitable for performing Grover's algorithm and has been numerically studied as a candidate for attacking NP-complete problems. More recently, deterministic random walks in continuous time have been proven to solve a classically hard problem in polynomial time [4]. Many other quantum algorithms can be mapped to the above families and, therefore, bring no further insight.

Some attempts to uncover an underlying principle, common to all known optimal algorithms, have already been explored though not definite and satisfactory answer has been found. One relevant instance is the role of entanglement in quantum algorithms [5–10]. Although entan-

glement is a natural resource to be exploited in quantum algorithm design, there are known examples of fast algorithms where the quantum register remains in a product state all along the computation. Our work will concentrate on quite a different proposal. The basic idea is that efficiency is related to a strong majorization principle. We shall investigate the way the probability distribution associated to the computational basis evolves along optimal quantum computations and find that it obeys a very constrained behavior we shall analyze in detail.

Let us recall that majorization theory arises as the natural framework to study the measure of disorder for classical probability distributions [11–14]. The notion of ordering emerging from majorization is far more severe than the one quantified by the standard Shannon entropy. If one probability distribution majorizes another, a set of inequalities must hold to constrain the former probabilities with respect to the latter. These inequalities lead to entropy ordering, but the converse is not necessarily true. In quantum mechanics, it has been proven that majorization is at the heart of the solution of a large number of quantum information problems. Majorization plays a fundamental role in topics like ensemble realization, conversion of quantum states via local operations and classical communication, and characterization of positive operator valued measurements [15].

In the context of quantum algorithms, a majorization principle has been formulated, proven for Grover's

^a e-mail: orus@ecm.ub.es

algorithm and verified for Shor's algorithm in reference [16]. Furthermore, a complete proof of majorization in quantum phase-estimation algorithms was presented in [17]. The underlying idea behind these analysis is that the probability distribution associated to the quantum state in the computational basis is step-by-step majorized until it is maximally ordered. Then a measurement will provide the sought solution with high probability. It has also been proven that the way such a detailed majorization emerges in both algorithmic families is intrinsically different [17].

In this paper we analyze the consistency and universality of a possible Majorization Principle. More concretely, we have studied several different quantum algorithms based on distinct properties. First, we have considered the problem of finding a hidden affine function by means of calls to an oracle. This problem is relevant because it is a known fast quantum algorithm that uses no entanglement at all. Second, we have taken a non-efficient instance, namely the parity determination problem. This is a case where the final solution must match a global majorization, yet it does not obey step by step majorization neither presents any quantum speed-up. The third case considered here is the large class of quantum adiabatic evolution algorithms. Efficiency and optimality has been proven to depend on the interpolating time path taken along the evolution. It is a remarkable fact that optimality appears when step-by-step majorization is present. Finally, deterministic quantum random walks provide exponential speed-up over classical oracle based random walks. Again, a manifest strong majorization arrow is detected.

The conclusion of the accumulated research is that all the analyzed instances of quantum algorithms support a step-by-step Majorization Principle. That is, optimal quantum computation is systematically verified to correspond to a step by step detailed reordering of the whole probability distribution in the computational basis. Some of the instances show the extra feature that the initial state can be prepared in different ways. Then, an initial step-by-step reverse majorization period precedes the subsequent step-by-step majorization, closing an invertible cycle. The study of quantum computation by adiabatic evolution shows the possibility of slower algorithms that maintain majorization. This implies that step-by-step majorization may be a necessary but is definitely not sufficient condition for efficiency.

We have structured the paper as follows: in Section 2 we briefly review some concepts about majorization theory and how it is related to quantum algorithms. We develop an analysis of a quantum algorithm for solving a hidden affine function problem in Section 3. In Section 4 we study an optimal quantum algorithm solving the parity problem. We move to an investigation of adiabatic quantum computation in Section 5, analyzing the effect of the evolution path in adiabatic searching algorithms in Sections 5.1 and 5.2, as well as the effect of the speed in the time variation of the Hamiltonian in Section 5.3. A further example of adiabatic evolution solving the 2-SAT "ring of agrees" problem is provided in Section 5.4. In Section 6

we examine a recently proposed quantum algorithm based on a continuous time quantum walk on a graph solving a classically hard problem. Finally, in Section 7 we state a Majorization Principle based on all the previous results and collect our conclusions.

2 Majorization theory and its relation to quantum algorithms

Our approach to the mathematical study of quantum processes will be through majorization's eye. We review in this section the contact point between majorization theory and quantum algorithms, as stated previously in [16,17]. In particular, we also present here the concept of "natural majorization", first stated in [17], which will eventually be used in this work.

2.1 Majorization theory

Let us consider two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ such that $\sum_{i=1}^d x_i = \sum_{i=1}^d y_i = 1$, whose components represent two different probability distributions. We say that distribution \mathbf{y} majorizes distribution \mathbf{x} , written as $\mathbf{x} \prec \mathbf{y}$, if and only if there exist a set of permutation matrices P_j and probabilities p_j such that

$$\mathbf{x} = \sum_j p_j P_j \mathbf{y}. \quad (1)$$

Because the probability distribution \mathbf{x} can be obtained from \mathbf{y} by means of a probabilistic sum, the definition given in equation (1) provides the intuitive notion that the \mathbf{x} -distribution is more disordered than \mathbf{y} . An alternative and usually more practical definition of majorization can be stated in terms of a set of inequalities to be held between the two distributions. Consider the components of the two vectors sorted in decreasing order, written as $(z_1, \dots, z_d) \equiv \mathbf{z}^\downarrow$. We say that \mathbf{y}^\downarrow majorizes \mathbf{x}^\downarrow if and only if the following relations are satisfied:

$$\sum_{i=1}^k x_i \leq \sum_{i=1}^k y_i \quad k = 1 \dots d. \quad (2)$$

In this paper we call probability sums similar to the ones appearing in the previous expression as "cumulants". There is still a third way of defining majorization involving the use of doubly stochastic matrices. A real $d \times d$ matrix $D = (D_{ij})$ is said to be doubly stochastic if it has non-negative entries, and each row and column of D sums to 1. We say that \mathbf{y} majorizes \mathbf{x} if and only if there is a doubly stochastic matrix D such that

$$\mathbf{x} = D\mathbf{y}. \quad (3)$$

The equivalence among the three given definitions can be proven [14]. Complementarily, we say that the probability distribution \mathbf{x} reversely majorizes distribution \mathbf{y} if and only if \mathbf{y} majorizes \mathbf{x} . A powerful relation involving majorization and the common Shannon entropy $H(\mathbf{x}) \equiv -\sum_{i=1}^d x_i \log x_i$ of probability distribution \mathbf{x} is that if $\mathbf{x} \prec \mathbf{y}$ then $H(\mathbf{x}) \geq H(\mathbf{y})$.

2.2 Link between majorization theory and quantum algorithms

The way we relate majorization theory to quantum algorithms can be stated as follows: let $|\psi^{(m)}\rangle$ be the pure state representing the register in a quantum computer at an operating stage labeled by $m = 1 \dots M$, where M is the total number of steps in the algorithm, and let N be the dimension of the Hilbert space. If we denote as $\{|i\rangle\}_{i=1}^N$ the basis in which the final measurement is performed in the algorithm, we can naturally associate a set of sorted probabilities p_i , $i = 1 \dots N$, to this quantum state in the following way: decompose the register state in the measurement basis such that

$$|\psi^{(m)}\rangle = \sum_{i=1}^N a_i^{(m)} |i\rangle. \quad (4)$$

The probability distribution associated to this state is

$$\mathbf{p}^{(m)} = \{p_i^{(m)}\} \quad p_i^{(m)} \equiv |a_i^{(m)}|^2 = |\langle i|\psi^{(m)}\rangle|^2 \quad (5)$$

where $i = 1 \dots N$, which corresponds to the probabilities of all the possible outcomes if the computation were stopped at stage m and a measurement were performed. A quantum algorithm will be said to majorize this probability distribution between steps m and $m+1$ if and only if [16, 17]

$$\mathbf{p}^{(m)} \prec \mathbf{p}^{(m+1)}. \quad (6)$$

Similarly, a quantum algorithm will be said to reversely majorize this probability distribution between steps m and $m+1$ if and only if

$$\mathbf{p}^{(m+1)} \prec \mathbf{p}^{(m)}. \quad (7)$$

If the situation given in equation (6) is step-by-step verified, there is a net flow of probability directed to the values of highest weight, in such a way that the probability distribution will be steeper as time flows. In physical terms, this can be stated as a very particular constructive interference behavior, namely, a constructive interference that has to step-by-step satisfy the constraints given in equation (2). The quantum algorithm builds up the solution at each time step by means of this very precise reordering of probability distribution.

It is important to note that majorization is checked on a particular basis. Step-by-step majorization is a basis dependent concept. Nevertheless there is a preferred basis, which is the basis defined by the physical implementation of the quantum computer or computational basis. The principle we analyze is rooted in the physical possibility to arbitrarily stop the computation at any time and perform a measurement. The claim pursued along the paper is that the probability distribution associated to this physically meaningful action obeys majorization.

2.3 Natural majorization in quantum algorithms

Let us now define the concept of natural majorization for quantum algorithms as it was originally presented in [17].

Working with probability amplitudes in the basis $\{|i\rangle\}_{i=1}^N$, the action of a particular unitary gate at step m makes the amplitudes evolve to step $m+1$ in the following way:

$$a_i^{(m+1)} = \sum_{j=1}^N U_{ij} a_j^{(m)}, \quad (8)$$

where U_{ij} are the matrix elements in the chosen basis of the unitary evolution operator (namely, the propagator from step m to step $m+1$). Inverting the evolution, we can write

$$a_i^{(m)} = \sum_{j=1}^N C_{ij} a_j^{(m+1)}, \quad (9)$$

where C_{ij} are the matrix elements of the inverse unitary evolution (which is unitary as well). Taking the square modulus we find

$$|a_i^{(m)}|^2 = \sum_{j=1}^N |C_{ij}|^2 |a_j^{(m+1)}|^2 + \text{interference terms.} \quad (10)$$

Should the interference terms disappear, majorization would be verified in a “natural” way between steps m and $m+1$ because the initial probability distribution could be obtained from the final one only by the action of a doubly stochastic matrix with entries $|C_{ij}|^2$. This is the so-called “natural majorization”: majorization which naturally emerges from the unitary evolution due to the lack of interference terms when making the square modulus of the probability amplitudes. Similarly, we can define the concept of “natural reverse majorization”, which follows in a similar way: there will be “natural reverse majorization” between steps m and $m+1$ if and only if there is “natural majorization” between time steps $m+1$ and m .

2.4 Majorization in Grover’s and Shor’s quantum algorithms

In order to motivate the forthcoming study we briefly sketch here the main results found concerning the analysis of majorization in the two well-known quantum algorithms of Grover [1] and Shor [2]. These results were already presented in references [16, 17], so we address the reader interested in precise details to these references.

On the one hand, Grover’s quantum searching algorithm was found in [16] to follow a step-by-step majorization. More concretely, each time the Grover’s operator is applied the probability distribution obtained from the computational basis obeys the set of constraints given in equation (2) until the searched state is found. Furthermore, because of the possibility of understanding Grover’s quantum evolution as a rotation in a two-dimensional Hilbert space (see for example [22]) it is seen that the quantum algorithm follows a step-by-step reverse majorization when evolving far away from the marked state, until the initial superposition of all possible computational states is obtained again. The quantum algorithm behaves such that majorization is present when approaching to the

solution, while reverse majorization appears when escaping from it. A cycle of majorization and reverse majorization emerges in the process as we consider long enough time evolutions, due to the rotational nature of Grover's operator.

On the other hand, Shor's quantum algorithm was analyzed inside of the broad family of quantum phase-estimation algorithms. In [16] it was observed that a step-by-step majorization seemed to appear under the action of the last quantum Fourier transform when considered in the usual Coppersmith decomposition [23]. One step further was taken in [17], where the complete mathematical proof of this property was provided. The result relies on the fact that those quantum states that can be mixed by a Hadamard operator coming from the decomposition of the quantum Fourier transform only differ by a phase all along the computation. Such a property entails as well the appearance of natural majorization, in the way presented above. Natural majorization was proven relevant for the case of Shor's quantum Fourier transform. This particular algorithm manages step-by-step majorization in a most efficient way. No interference terms spoil the majorization introduced by the natural diagonal terms in the unitary evolution. It is still unclear the role that natural majorization plays in distinguishing different levels of efficiency in quantum algorithms.

These two results suggest a possible relation between majorization and quantum algorithms. This is the point to be exploited in detail with further examples in the next sections.

3 Analysis of a quantum algorithm for solving a hidden affine function problem

The problem of finding hidden affine functions was initially proposed by Bernstein and Vazirani [18] as a generalization of Deutsch's problem [19]. Further studies have investigated into this class of problems, providing a range of fast quantum algorithms for solving different generalizations [20,21]. The case we present in this work is one of the multiple variations stated in reference [21], but the main results we obtain can be verified as well for the whole set of quantum algorithms solving similar problems.

3.1 Setting of the problem

Let us state the following problem (see [21]): *given an integer N function $f: x \rightarrow mx + b$, where $x, m, b \in \mathbb{Z}_N$, find m .*

The classical analysis reveals that no information about m can be obtained with only one evaluation of the function f . Conversely, given the unitary operator U_f acting in a reversible way in the Hilbert space $\mathbb{H}_N \otimes \mathbb{H}_N$ such that

$$U_f|x\rangle|y\rangle = |x\rangle|y + f(x)\rangle, \quad (11)$$

(where the sum is to be interpreted as modulus N), there is a quantum algorithm solving this problem with only one query to U_f .

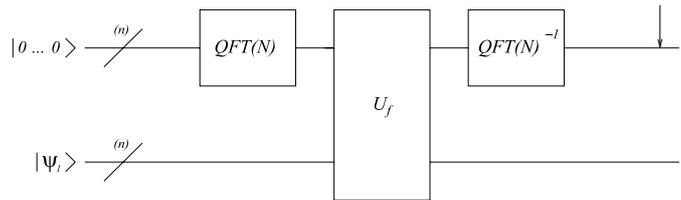


Fig. 1. Quantum circuit solving the hidden affine function problem. Each quantum wire is assumed to be composed of n qubits. The arrow at the end indicates a measurement.

3.2 Quantum algorithm

Let us take $N = 2^n$, being n the number of qubits. The quantum algorithm optimally solving the problem previously presented reads as follows:

- prepare two registers of n qubits in the state $|0 \dots 0\rangle|\psi_1\rangle \in \mathbb{H}_N \otimes \mathbb{H}_N$, where $|\psi_1\rangle = QFT(N)^{-1}|1\rangle$, and $QFT(N)^{-1}$ denotes the inverse quantum Fourier transform in a Hilbert space of dimension N ;
- apply $QFT(N)$ over the first register;
- apply U_f over the whole quantum state;
- apply $QFT(N)^{-1}$ over the first register;
- measure the first register and output the measured value.

The different steps concerning this process are summarized in Figure 1.

3.3 Analysis of the quantum algorithm

We now show how the proposed quantum algorithm leads to the solution of the problem. Our analysis raises two observations concerning the way both entanglement and majorization behave in the computational process.

In the first step of the algorithm, the quantum state is separable, noting that the quantum Fourier transform (and its inverse) applied on a well defined state in the computational basis leads to a perfectly separable state (see, for example, [20]). Actually, this separability holds also step-by-step when a decomposition for the quantum Fourier transform is considered, such as the Coppersmith's decomposition [23]. That is, the quantum state $|0 \dots 0\rangle|\psi_1\rangle$ is unentangled.

The second step of the algorithm corresponds to a quantum Fourier transform in the first register. This action leads to a step-by-step reverse majorization of the probability distribution of the possible outcomes while it does not use neither create any entanglement. Moreover, natural reverse majorization is at work due to the absence of interference terms.

Next, it is easy to verify that the quantum state

$$|\psi_1\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{-2\pi i \frac{j}{N}} |j\rangle \quad (12)$$

is an eigenstate of the operation $|y\rangle \rightarrow |y + f(x)\rangle$ with eigenvalue $e^{2\pi i \frac{f(x)}{N}}$. After the third step, the quantum

state reads

$$\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} e^{2\pi i \frac{f(x)}{N}} |x\rangle |\psi_1\rangle = \frac{e^{2\pi i \frac{b}{N}}}{\sqrt{N}} \left(\sum_{x=0}^{N-1} e^{2\pi i \frac{ax}{N}} |x\rangle \right) |\psi_1\rangle. \quad (13)$$

The probability distribution of possible outcomes has not been modified, thus not affecting majorization. Furthermore, the pure quantum state of the first register can be written as $QFT(N)|m\rangle$ (up to a phase factor), so this step has not created any entanglement among the qubits of the system either.

In the fourth step of the algorithm, the action of the operator $QFT(N)^{-1}$ over the first register leads to the state $e^{2\pi i \frac{b}{N}} |m\rangle |\psi_1\rangle$. A subsequent measurement in the computational basis over the first register provides the desired solution. Recalling the results found in [17], we see that the inverse quantum Fourier transform naturally majorizes step-by-step the probability distribution attached to the different outputs. On the other hand, the separability of the quantum state still holds step-by-step. Note that step-by-step majorization is in fact dependent on the specific implementation of the quantum Fourier transform operation, but nevertheless it holds true for other decompositions of the operator apart from the usual Coppersmith's one (as we already stated in [17]).

It is clear that the quantum algorithm is faster than any of its possible classical counterparts, as it only needs of a single query to the unitary operator U_f to get the solution. We can summarize our analysis of majorization for the present quantum algorithm as follows.

Result 1. *The fast quantum algorithm for finding a hidden affine function shows a majorization cycle based on the action of a $QFT(N)$ and a $QFT(N)^{-1}$.*

We understand that the algorithm is entanglement-free as long as we analyze it between the action of the different unitary gates. From a more physical point of view, the quantum registers may become highly entangled during the performance of multi-qubit gates, despite it is not present between two of them. Our assertion relies then on the study of the system at these particular steps in the computation, which we think to be the most natural steps to consider. It follows that there can exist quantum computational speed-up without the use of entanglement (in the way made precise before). In this case, it is seen that no resource increases exponentially. Yet, a majorization cycle is present in the process, which is indeed rooted in the structure of both the quantum Fourier transform and the quantum state.

4 Analysis of an optimal quantum algorithm for solving the parity problem

The problem of finding the parity of a given function $f: x \in \mathbb{Z}_N \rightarrow \{-1, +1\}$, usually known as the parity problem, has been shown to be as hard for a quantum computer in the quantum oracular model of computation as it is for a classical computer [24,25]: no quantum speed-up can be

achieved in this case. We shall first present the problem and then analyze an optimal quantum algorithm proposed in [24] from the point of view of majorization.

4.1 Setting of the problem

Let us state the parity problem in the following way (see [24]): *given a function $f: x \in \mathbb{Z}_N \rightarrow \{-1, +1\}$, evaluate the product of all the $f(x)$ for all the possible x .*

It has been proved that a quantum computer will need at least $N/2$ queries to the quantum oracle f for solving this problem compared to the N classical queries [24]. Thus, a quantum computer is not faster than a classical one (in the limit of a very large input, where N goes to infinity) when dealing with this situation: no better efficiencies can be obtained using the quantum computational paradigm in getting the desired result. The time complexity of the best possible quantum algorithm will be $O(N)$, without improvement with respect to the classical time complexity, because the quantum speed-up is only by a factor of two.

4.2 Quantum algorithm

Let us briefly outline the main lines of an optimal quantum algorithm solving the parity problem in exactly $N/2$ queries to the oracular function f , which was initially presented in [24]. We first introduce a series of definitions and notations: the function $f(x)$ will be evaluated through a quantum oracle acting in the following way

$$\begin{aligned} U_f |x, +1\rangle &= |x, f(x)\rangle \\ U_f |x, -1\rangle &= |x, -f(x)\rangle, \end{aligned} \quad (14)$$

where the second register is a qubit taking the values ± 1 . Let us define also the quantum state $|x, a\rangle = (|x, +1\rangle - |x, -1\rangle) / \sqrt{2}$, which is seen to be a proper state of operator U_f with eigenvalue $f(x)$. With these definitions, the quantum algorithm reads as follows:

- prepare the initial quantum state $|\psi_0\rangle = \sum_{x=1}^N |x, a\rangle / \sqrt{N}$;
- apply the following operations over the initial quantum state:

$$V_{N/2} U_f V_{N/2-1} U_f \cdots V_1 U_f |\psi_0\rangle \equiv |\psi_f\rangle, \quad (15)$$

where U_f is defined as before, and the rest of operators are defined as $V_{N/2} = 1$, $V_i = V \forall i \neq N/2$, with

$$\begin{aligned} V|x, a\rangle &= |x+1, a\rangle \quad x = 1, \dots, \frac{N}{2} - 1 \\ V|\frac{N}{2}, a\rangle &= |1, a\rangle \\ V|x, a\rangle &= |x+1, a\rangle \quad x = \frac{N}{2} + 1, \dots, N-1 \\ V|N, a\rangle &= |\frac{N}{2} + 1, a\rangle; \end{aligned} \quad (16)$$

- measure the observable $|\psi_0\rangle\langle\psi_0|$ over $|\psi_f\rangle$.

Note that the final measurement is to be made on a specific basis.

4.3 Analysis of the quantum algorithm

We now make an study of how this algorithm leads to the desired solution of the proposed problem. The analysis will show the way majorization behaves in this optimal but non-efficient quantum process.

We focus on how the operations in the second step affect the quantum state leading to the solution, and thus affect the probability distribution of possible outcomes for the final measurement. If we apply U_f to the initial state, we get

$$U_f|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{x=1}^N f(x)|x, a\rangle. \quad (17)$$

After the application of operator V_1 the quantum state evolves to

$$\begin{aligned} V_1 U_f |\psi_0\rangle &= \frac{1}{\sqrt{N}} \sum_{x=1}^{N/2} f(x-1)|x, a\rangle \\ &+ \frac{1}{\sqrt{N}} \sum_{x=N/2+1}^N f(x-1)|x, a\rangle, \end{aligned} \quad (18)$$

as can be directly checked (care must be taken with the possible values of x in both sums). If we now apply again U_f we get:

$$\begin{aligned} U_f V_1 U_f |\psi_0\rangle &= \frac{1}{\sqrt{N}} \sum_{x=1}^{N/2} f(x)f(x-1)|x, a\rangle \\ &+ \frac{1}{\sqrt{N}} \sum_{x=N/2+1}^N f(x)f(x-1)|x, a\rangle, \end{aligned} \quad (19)$$

so we begin to recognize the pattern the algorithm follows. At the end of the computation the final state is

$$\begin{aligned} |\psi_f\rangle &= \frac{1}{\sqrt{N}} f(1) \cdots f(N/2) \sum_{x=1}^{N/2} |x, a\rangle \\ &+ \frac{1}{\sqrt{N}} f(N/2+1) \cdots f(N) \sum_{x=N/2+1}^N |x, a\rangle, \end{aligned} \quad (20)$$

and it is easily verified that this state is equal to $|\psi_0\rangle$ if the parity of the function is equal to $+1$, and orthogonal to $|\psi_0\rangle$ (say $|\psi_0^\perp\rangle$) if the parity is -1 . A suitable measurement of the observable $|\psi_0\rangle\langle\psi_0|$ can then distinguish between the two values.

Let us now analyze the way this algorithm behaves with respect to majorization. As stated in Section 2.2, majorization must always be checked from the probability distribution of obtaining the different outcomes of the

final measurement. In other words, the probability distribution subject of analysis must always be the one obtained from the final measurement basis. Such a basis turns usually to be the computational one, but it is not necessarily so. Here, we are dealing with one of these exceptional cases in which the final measurement basis differs from the computational one. Consequently, majorization must be studied in this unusual but natural basis for the quantum algorithm.

The only two vectors we know of this basis are $|\psi_0\rangle$ and $|\psi_0^\perp\rangle$. We could extend them to a whole basis but it is not necessary for our purposes, as we can analyze the probability of being in each of these two states all along the computation. Should majorization be present step-by-step in the process, the probability of being in one of these two states would smoothly decrease in favor of the other one, which would parallelly smoothly increase. In a naive way, this is what a *majorization arrow* means: as the process evolves the probability of being in the right state becomes bigger and bigger.

We can observe that this does not happen in the algorithm for the whole computational process, except for the last application of the oracle U_f (compulsory if one wishes to distinguish between the two states). It is easily seen, because in all the steps of the computation the quantum state is an arbitrary superposition of computational states of amplitudes $+1/\sqrt{N}$ and $-1/\sqrt{N}$, without any apparent structure, so the probabilities of being in $|\psi_0\rangle$ or $|\psi_0^\perp\rangle$ evolve erratically. The full structure only appears when the last oracle U_f is applied, thus providing the necessary majorization to be able to distinguish the two states with a measurement, but the important point is that this is not a majorization arrow, because there is no step-by-step majorization. This is stated in our second result.

Result 2. *No step-by-step majorization is present along the non-efficient parity determination problem.*

Thus, no majorization cycle similar to the one found in the preceding section could ever appear. Interestingly enough, this is a problem in which quantum computers do not provide a better efficiency than classical ones.

5 Analysis of the adiabatic searching algorithms and of an adiabatic algorithm solving a 2-SAT problem

We now turn to the analysis of the quantum adiabatic searching algorithm, observing the effects of a change of path between the initial and the problem Hamiltonian under the majorization's point of view [16]. We see that those paths leading to optimality in the algorithm lead as well to step-by-step majorization, while the converse is not necessarily true. A different example of adiabatic evolution is analyzed in the last point of this section, namely, an adiabatic algorithm solving the 2-SAT "ring of agrees" problem.

The adiabatic model of quantum computation was initially introduced in [3], and can be briefly summarized as follows. We consider a physical system controlled by a time dependent Hamiltonian of the form

$$H(s(t)) = (1 - s(t))H_0 + s(t)H_p, \quad (21)$$

where H_0 and H_p are the initial and problem Hamiltonian respectively, and $s(t)$ is a time dependent function satisfying the boundary conditions $s(0) = 0$ and $s(T) = 1$ for a given T . The desired solution to a given problem is encoded in the ground state of H_p . The gap between the ground and the first excited state of the instantaneous Hamiltonian at time t will be called $g(t)$. Let us define g_{min} as the global minimum of $g(t)$ for t in the interval $[0, T]$. If at time T the ground state is given by the state $|E_0; T\rangle$, the adiabatic theorem states that if we prepare the system in its ground state at $t = 0$ (which is assumed to be easy to prepare) and let it evolve under this Hamiltonian, then

$$|\langle E_0; T | \psi(T) \rangle|^2 \geq 1 - \epsilon^2 \quad (22)$$

provided that

$$\frac{\max_t \left| \frac{dH_{1,0}}{dt} \right|}{g_{min}^2} \leq \epsilon \quad (23)$$

where $H_{1,0}$ is the Hamiltonian matrix element between the ground and first excited state, $\epsilon \ll 1$, and the maximization is taken over the whole time interval $[0, T]$ [3, 28]. Because the problem Hamiltonian encodes the solution to the problem in its ground state, we get the desired solution with high probability after a time T .

This quantum adiabatic evolution method has been applied to the searching problem ([26–29]). Let the initial state be $|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{x=1}^N |x\rangle$, being N the number of entries for the searching, and let the initial and problem Hamiltonian be $H_0 = I - |\psi_0\rangle\langle\psi_0|$ and $H_p = I - |m\rangle\langle m|$, being $|m\rangle$ the searched state. This scheme leads to different results depending on whether we apply the adiabatic condition globally (that is, in the whole time interval $[0, T]$) or locally (at each time t). In what follows we will consider these two situations without entering into precise details of the involved calculations. For further information, we refer the reader to [26, 27] and references therein.

5.1 Analysis of the fastest global adiabatic evolution

Let us suppose that we demand the adiabatic condition given in equation (23) to be satisfied globally in the whole interval $[0, T]$. This does not involve any particular restriction on the form of $s(t)$, so we can then choose $s(t) = t/T$, leading to a linear evolution of the Hamiltonian. Under these circumstances it can be proven [27] that the global adiabatic condition is verified provided that

$$T \geq \frac{N}{\epsilon}. \quad (24)$$

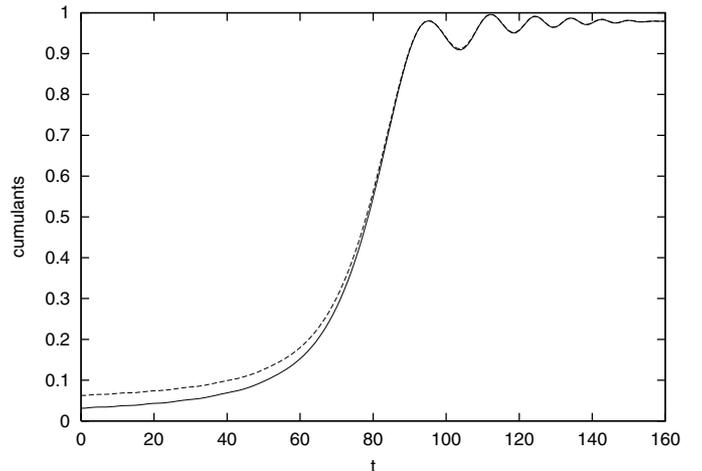


Fig. 2. Global adiabatic evolution with parameters $\epsilon = 0.2$, $N = 32$ and $T = 160$. The solid line corresponds to the time evolution of $P_+(t)$ and the dashed line that of $P_+(t) + P_-(t)$.

Hence, the algorithm needs $O(N)$ time to hit the solution with appreciable probability, so the global adiabatic searching does not lead to an increasing efficiency with respect to a classical searching (in contrast with the square root speed-up of Grover’s quantum algorithm).

Let us call $P_+(t)$ the probability of being at the searched state at time t and similarly $P_-(t)$ the probability of being at any different state from the desired one at time t . Note that, because of the symmetry of the problem, $P_-(t)$ will be exactly the same for all those quantum states differing from the solution state all along the computational process. In order to analyze majorization, we recall the set of inequalities given in equation (2) to be satisfied at each majorizing time step. It is easy to see that the maximum probability at all times will be $P_+(t)$, while the rest of the probabilities will remain smaller to this quantity all along the computation and equal to $P_-(t)$. In order to gain simplicity we have analyzed in detail the behaviour of the two non-trivial cumulants $P_+(t)$ and $P_+(t) + P_-(t)$, as the rest of them will not lead to different conclusions from the ones emerging from our study.

We have performed numerical simulations in the fastest allowed case ($T = N/\epsilon$) and have found the time evolution for the two cumulants. The results for $\epsilon = 0.2$ and $N = 32$ are shown in Figure 2.

From our numerical analysis we arrive at our third result.

Result 3. *A naive adiabatic quantum searching process does not produce an optimal algorithm neither verifies step-by-step majorization.*

This property is clearly seen as the two cumulants decrease in time for some time steps, thus not verifying majorization.

5.2 Analysis of the local adiabatic evolution

The preceding global adiabatic method can be improved if we apply the adiabatic condition given in equation (23)

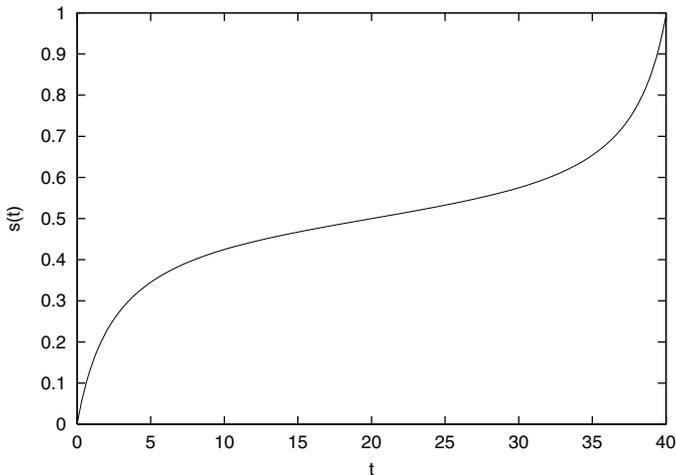


Fig. 3. Local adiabatic evolution. The driving Hamiltonian, with $\epsilon = 0.2$ and $N = 32$.

locally. That is, let us divide the interval $[0, T]$ into many subintervals and let us apply (23) to each of the subintervals individually. Taking the limit of the size of the subintervals going to zero, we find that the adiabatic restriction has to be fulfilled locally at each time t :

$$\left| \frac{dH_{1,0}}{dt} \right| \leq \epsilon \quad \forall t. \quad (25)$$

This is a less demanding condition than (23) (if (23) is satisfied, so is (25), but the inverse is not necessarily true), and means that the adiabatic evolution must be satisfied at each infinitesimal time interval. It can be shown (see, for example, [27]) that proceeding in this way the function $s(t)$ must have a precise form which is given by the relation

$$t = \frac{1}{2\epsilon} \frac{N}{\sqrt{N-1}} \left(\arctan(\sqrt{N-1}(2s-1)) + \arctan(\sqrt{N-1}) \right). \quad (26)$$

We can observe this dynamic evolution from Figure 3, in the case of $\epsilon = 0.2$ and $N = 32$. The local adiabatic process implies that the smaller the energy gap between the ground and first excited states, the slower the evolution of the Hamiltonian (and vice versa).

With this information it can be proven as well [27] that the evolution time for the algorithm to succeed with appreciable probability is, in the limit where $N \gg 1$,

$$T = \frac{\pi}{2\epsilon} \sqrt{N}. \quad (27)$$

Hence, in the case of local adiabatic evolution the computational process takes $O(\sqrt{N})$ time, just as in Grover's quantum searching algorithm, obtaining an square root speed-up with respect to a classical searching.

Defining again $P_+(t)$ and $P_-(t)$ in the same way as in Section 5.1, we can restrict ourselves to the analysis of the two non-trivial cumulants $P_+(t)$ and $P_+(t) + P_-(t)$ in

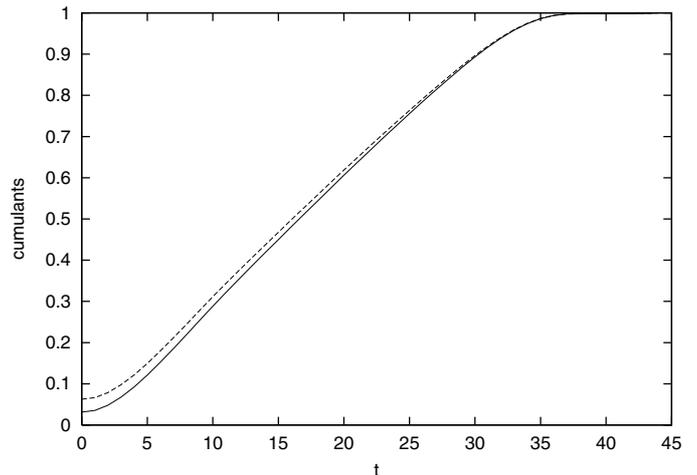


Fig. 4. Local adiabatic evolution with parameters $\epsilon = 0.2$, $N = 32$ and $T = 44$. The solid line corresponds to the time evolution of $P_+(t)$ and the dashed line that of $P_+(t) + P_-(t)$.

order to observe the evolution of majorization in the quantum process. We have numerically solved the equations for $\epsilon = 0.2$ and $N = 32$, and have found the evolution of the two quantities, which is given in Figure 4.

From the numerical analysis the following result emerges.

Result 4. *A local adiabatic searching algorithm is optimal and verifies step-by-step majorization.*

This result can be straightforwardly verified since the set of inequalities of (2) are satisfied step-by-step, according to the plot in Figure 4. Due to this behavior, *the whole computational process might lead to a majorization cycle*, such as the one observed in Section 3, as long as the previous preparation of the initial quantum state of the computation at time $t = 0$ leads to a step-by-step reverse majorization. This turns to be always possible, for example, by applying a set of Hadamard gates to the log N -qubit quantum state $|0 \dots 0\rangle$ (assume that N is a power of 2, for simplicity), which would efficiently prepare a superposition of all the possible quantum states together linked to a reverse majorization arrow. Furthermore, this quantum process leads to an increasing efficiency with respect to a classical searching, exactly in the same fashion as Grover's original quantum searching algorithm [1].

5.3 Analysis of slower global adiabatic evolutions

Let us now address the situation of global adiabatic evolutions which are not necessarily tight in time, that is, extremely slow time variations for the Hamiltonian, much slower than the minimum necessary for the adiabatic theorem to hold. In the case we are dealing with, that implies the consideration of the case in which $T > N/\epsilon$, i.e., the adiabatic inequality (24) is not tight. This case is not very relevant from a computational point of view because the hitting time is not the minimum possible, but we think it is worthwhile to be studied also from the point of view

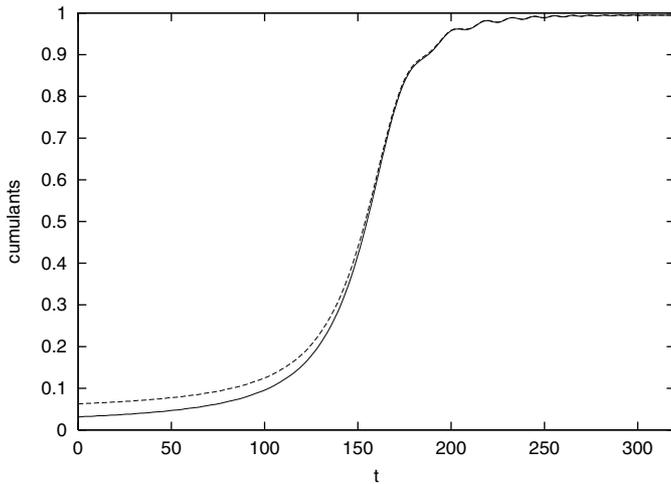


Fig. 5. Global adiabatic evolution with $\epsilon = 0.2$, $N = 32$, and $T = 320$. The solid line corresponds to the time evolution of $P_+(t)$ and the dashed line that of $P_+(t) + P_-(t)$.

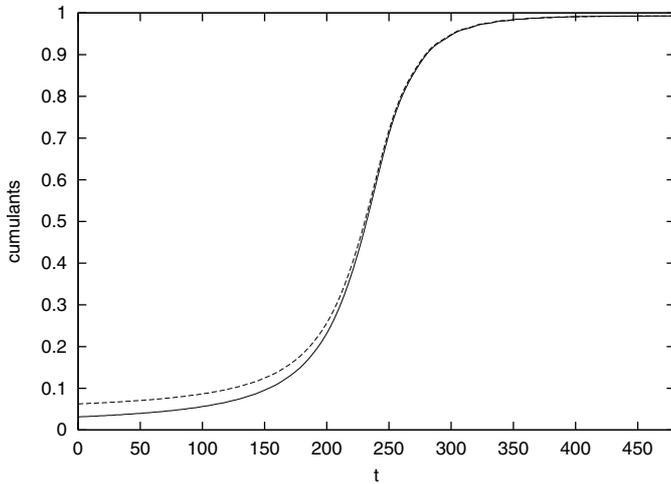


Fig. 6. Global adiabatic evolution with $\epsilon = 0.2$, $N = 32$, and $T = 480$. The solid line corresponds to the time evolution of $P_+(t)$ and the dashed line that of $P_+(t) + P_-(t)$.

of majorization theory in order to have a more complete picture of how majorization really works in this kind of quantum algorithms.

We have performed again numerical analysis for the time evolution of the two non-trivial cumulants, for $\epsilon = 0.2$, $N = 32$, and $T = 320, 480$ (in both cases bigger than $N/\epsilon = 160$). The results are plotted in Figures 5 and 6.

From these two plots, we observe with Figure 2, that a step-by-step majorization appears as long as the evolution of the Hamiltonian becomes slower and slower. Physically, this means that the probability of “jumping” to the first excited state diminishes as long as the evolution is performed with a very small velocity, thus satisfying better the assumptions of the adiabatic theorem. This leads to processes in which there is no quantum speed-up but there is indeed a majorization arrow. However, these processes are not optimal in time, as we can always find faster quan-

tum algorithms for solving the problem. Consequently, we arrive at the following conclusion.

Result 5. *Non-optimal quantum algorithms may present step-by-step majorization. In particular, step-by-step majorization may appear in global adiabatic searching processes for a slow enough evolution rate.*

It follows that step-by-step majorization cannot be a sufficient condition for quantum speed-up.

We can get some further intuition of the set of results presented in this section. Adiabatic quantum searching algorithms can be understood (in the limit of large N) as a rotation from the initial state to the marked state as long as the adiabaticity (either global or local) of the evolution is conveniently satisfied (see [29] for details). The difference between the global and local conditions turns out to be the evolution rate of the rotation angle: local adiabatic evolution imposes a rotation at constant rate (as in the original Grover’s algorithm) whereas global adiabatic evolution does not. Because of this rotational picture, step-by-step majorization is verified as long as the quantum state remains in the instantaneous ground state all along the computation. We can now understand our results in a finer way. Global adiabatic evolution is not a strong enough condition for adiabaticity, thus we only see step-by-step majorization when the evolution is *really* slow, in which case the quantum state adiabatically rotates towards the solution because it remains very close to the ground state of the instantaneous Hamiltonian. Local adiabatic evolution is a stronger condition for adiabaticity the quantum state remains always very close to the instantaneous ground state, thus performing the rotation towards the solution which in turn involves step-by-step majorization.

5.4 A further example: a 2-SAT quantum adiabatic algorithm solving the “ring of agrees” problem

Let us consider now a different example of adiabatic quantum computation, namely, an adiabatic quantum algorithm solving the 2-SAT “ring of agrees” problem, as stated in [3]. As long as 2-SAT can be efficiently solved by a classical algorithm in a time $O(\text{poly}(n))$ (being n the number of bits) [30], quantum computation can do no better than classical computation in this case. The problem Hamiltonian H_p is now a sum of Hamiltonians involving each of the different clauses of the 2-SAT problem, whereas the initial Hamiltonian H_0 is such that its ground state is again an equal superposition of all the possible states of the computational basis. The “ring of agrees” problem over n bits is defined such that clause j acts on bits j and $j + 1$ where j runs from 1 to n and bit $n + 1$ is identified with bit 1. Each clause is an “agree” clause, which means that 00 and 11 are the satisfying assignments. The eigenvectors of the Hamiltonian associated with clause j are the computational states, in such a way that those which “agree” in qubits j and $j + 1$ have zero energy (ground states) whereas those which “disagree” have energy one. Because the problem Hamiltonian is a sum of the

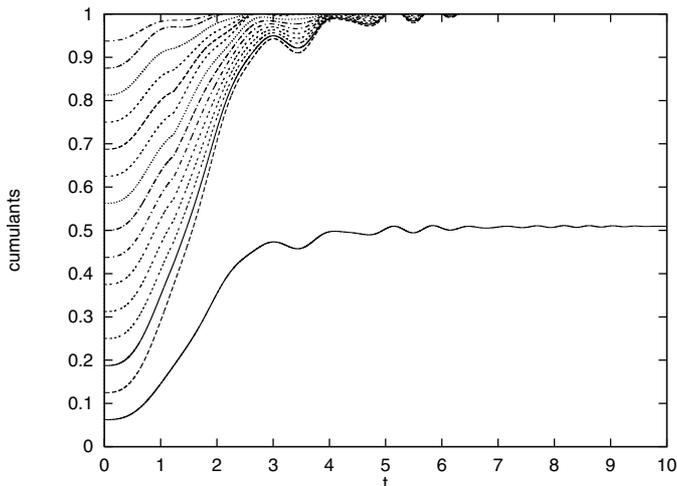


Fig. 7. Evolution of the 15 cumulants in the “ring of agrees” for a Hilbert space of dimension 16.

n Hamiltonians for the n “agree” clauses, its ground states are $|0\rangle|0\rangle \cdots |0\rangle$, $|1\rangle|1\rangle \cdots |1\rangle$, or any linear combination of them.

We have made an analysis of the adiabatic quantum algorithm in which the interpolation between the initial and problem Hamiltonian is linear, $s(t) = t/T$, for the case of 4 qubits (Hilbert space of dimension 16), and choosing $T = 10$. The evolution of the 15 cumulants is plotted in Figure 7.

Note the similarity between Figures 7 and 2 in Section 5.1, namely, no step-by-step majorization is present in the evolution because of the oscillatory behaviour of the cumulants. Both plots represent quantum algorithms which do not improve classical computation and which probabilities share the same behaviour under the point of view of majorization. Our observation is then the following.

Result 6. *A quantum adiabatic algorithm solving the 2-SAT “ring of agrees” problem does not improve classical computation, neither verifies step-by-step majorization.*

This result reinforces the ones already found with respect to adiabatic searching algorithms.

6 Analysis of a quantum walk in continuous time with exponential algorithmic speed-up

The extension of classical random walks to the quantum world has been widely studied, yielding two different models of quantum random walks, namely, those which operate in discrete time by means of using a “coin operator” [31–33] and those based on a Hamiltonian evolution in continuous time [4, 34, 35]. Regarding the discrete time model of quantum random walk two interesting algorithmic results have been found so far, namely, an exponentially faster hitting time in the hypercube with respect to the classical random walk [36] and a quantum searching algorithm achieving the Grover’s quadratic speed-up [37].

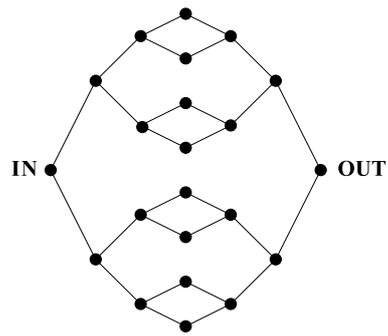


Fig. 8. A possible graph constructed from two binary trees with $n = 3$.

The first of these examples does not provide any algorithmic speed-up, as there exists a classical algorithm that solves the hitting problem in the hypercube exponentially faster than the naive classical random walk, that is, in a time $O(\text{poly}(\log N))$ where N is the number of nodes of the graph (see [4]). On the other hand, the second of these examples shows algorithmic advantage with respect to any possible classical computation. The analysis of the quantum random walk searching algorithm shows that the quantum evolution can be understood as an (approximate) rotation of the quantum state in a two-dimensional Hilbert space which is exact in the limit of a very large database (see [37] for details), resembling the original proposal of Grover’s searching algorithm which can be decomposed exactly in a two-dimensional Hilbert space. This rotational structure of the evolution implies step-by-step majorization when approaching the marked state, exactly in the same fashion than the usual Grover’s searching algorithm (already analyzed in [16]).

In this section we restrict ourselves to the continuous time model of quantum walk and analyze a recently proposed quantum algorithm based on a quantum walk on continuous time solving a classically hard problem [4]. Here we restrict ourselves to briefly sketch the main points and ideas of both the problem setting and its quantum efficient solution, since the whole development of the algorithm is not the purpose of the present paper. We address the interested reader to [4].

6.1 Setting of the problem

The problem we wish to solve is defined through a graph built in the following way (see [4]): suppose we are given two balanced binary trees of height n with the 2^n leaves of the left tree identified with the 2^n leaves of the right tree in a simple way, as shown in Figure 8. A way of modifying such a graph is by connecting the leaves by a random cycle that alternates between the leaves of the two trees, instead of identifying them directly. An example of such a graph is provided in Figure 9.

Suppose that the edges of such a graph are assigned a consistent coloring (that is, not two edges incident in the same vertex have the same color), and that the vertices are each one given a different name (with a $2n$ -bit string,

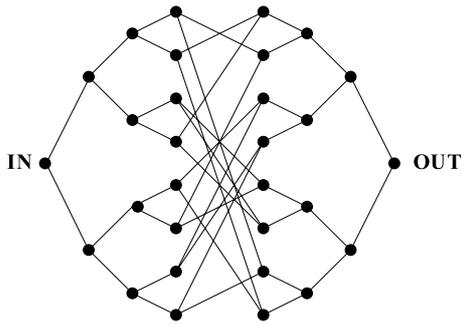


Fig. 9. An alternative graph constructed from two binary trees with $n = 3$ (connection between the leaves is made through a random cycle).

so there are more possible names than the ones assigned). We now define a black box that takes two inputs, a name a given as a $2n$ -bit string and a color c , and acts in the following way: if the input name a corresponds to a vertex that is incident with an edge of color c , then the output corresponds to the name of the vertex joined by that edge; if a is not the name of a vertex or a is the name of a vertex but there is no incident edge of color c , the output is the special $2n$ -bit string $11 \dots 1$, which is not the name of any vertex.

Now, the problem we wish to solve reads as follows: given a black box for a graph such as the one previously described, and given the name of the IN vertex, find the name of the OUT vertex.

In [4] it was proven that no classical algorithm can transverse a graph such as the one in Figure 9 in polynomial time, given such a black box. Furthermore, an explicit construction of a quantum algorithm based on a continuous time quantum walk on the graph that succeeds in finding the solution for this oracular problem in polynomial time was given.

6.2 Quantum algorithm

The quantum algorithm of [4] can be briefly summarized as follows: consider the $(2n + 2)$ -dimensional subspace spanned by the states

$$|\text{col } j\rangle = \frac{1}{\sqrt{N_j}} \sum_{a \in \text{column } j} |a\rangle, \quad (28)$$

where $N_j = 2^j$ if $0 \leq j \leq n$ and $N_j = 2^{2n+1-j}$ if $n+1 \leq j \leq 2n+1$. We will call this subspace the “column subspace”, and each state of the basis is an equally weighted sum of the states corresponding to the vertices lying on each column of the graph. We now define a Hamiltonian acting on this subspace in the following way:

$$\langle \text{col } j | H | \text{col } (j+1) \rangle = \begin{cases} 1 & 0 \leq j \leq n-1, \quad n+1 \leq j \leq 2n \\ \sqrt{2} & j = n \end{cases} \quad (29)$$

with hermiticity of H giving the other nonzero matrix elements. The action of this Hamiltonian in the graph is

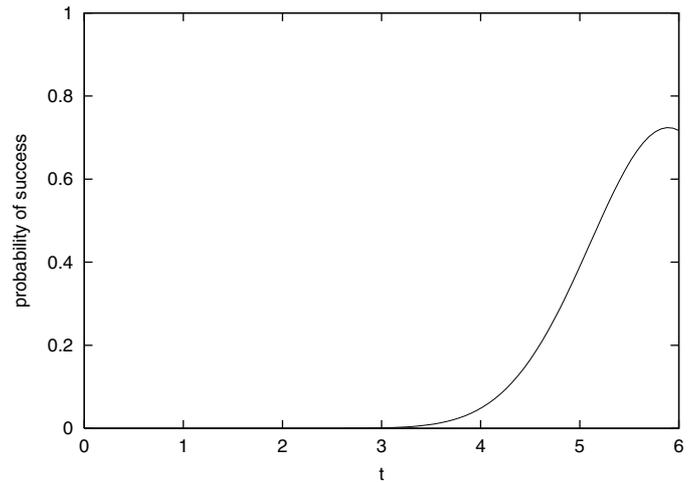


Fig. 10. Quantum walk algorithm. Probability of finding the exit, for $n = 4$.

nothing but promoting transitions between adjoint vertices, so a quantum walk on the graph (on the whole Hilbert space) generated by this Hamiltonian is equivalent to a quantum walk on the line (on the column subspace). Consequently, from now on we only focus our attention in the quantum walk in the line generated by the Hamiltonian of (29). Moreover, it can be proven that given the structure of the graph in the form of a black box such as the one already described, our Hamiltonian can be efficiently simulated [4].

The quantum walk works as follows: at first the “wave packet” will be precisely localized at the IN vertex (the initial state will be $|\text{col } 0\rangle$). Due to unitary time evolution, it will initially spread out through the different vertices at the left hand side of the graph (those belonging to the left binary tree), but after a short time (once half the graph has been transversed) it will begin to spread through the vertices on the right hand side, interfering constructively in the OUT vertex as the time goes on. Physically, this is nothing but a wave propagation. Should we wait more time, the wave packet would come back to the entrance, and the process would be similarly repeated again. Actually, due to the “defect” of the Hamiltonian in the middle vertices, it can be shown that the transmission through the central columns is not of 100 percent (thus providing interferences in long enough time scales), but high enough for the OUT node to be achieved with a very high probability. In [4] the authors prove that the succeeding time is polynomial in n .

6.3 Analysis of the quantum algorithm

We have numerically simulated this quantum walk for the particular case of $n = 4$, and have plotted the time evolution of the probability of success in Figure 10. We observe that the numerical result fits with the prediction that the time the algorithm takes in achieving the OUT node is polynomial.

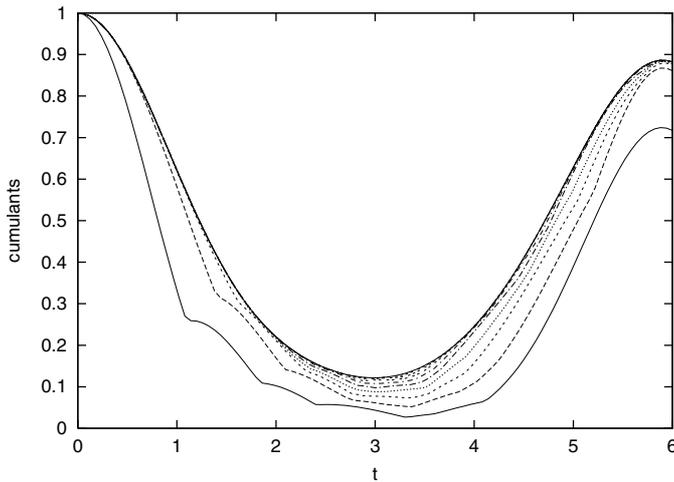


Fig. 11. Quantum walk algorithm. Time evolution of the ten cumulants when one node per column is considered, for $n = 4$.

It is easy to observe that, in order to analyze majorization, for the case of $n = 4$ there are only 10 non-trivial probabilities to be studied. This is so due to the fact that all the states of the whole Hilbert space belonging to the same column always share the same probability amplitude. The relevant quantities to be studied are then the probabilities of being at each column state normalized to the number of nodes belonging to that column, that is, the probability of being in one node of each column. There are then $2n + 2$ different probabilities to be considered at each time step. Given only this 10 quantities, we are able to calculate the whole set of 62 cumulants corresponding to all the sums of sorted probabilities, according with equation (2). In order to make the figures as clear as possible we only plot 10 of these quantities in Figure 11, corresponding to the cumulants arising from the sorted probabilities when only one node per column is considered. The rest of the cumulants can be shown to have a similar behavior to that of the ones appearing in Figure 11.

We have also numerically simulated the algorithm in the case of a bigger graph, namely, in the case $n = 10$. In this case there are $2n + 2 = 22$ different probabilities to be considered at each time step. Proceeding in the same way than in the case $n = 4$ (that is, not plotting all the cumulants, but the only the sorted sum of these 22 probabilities), we obtain a similar behaviour as in the case for $n = 4$, as is shown in Figure 12.

Therefore, we arrive at the following conclusion.

Result 7. *The continuous time quantum walk transversing a classically hard graph follows a step-by-step majorization cycle all along the computation until it reaches the OUT node.*

It is worth remarking as well that the time the algorithm spends reversely majorizing the probability distribution is about half of the time of the whole computation. The physical reason for this behavior is clear, as this is the time the “wave packet” spends spreading over the binary tree on the left hand side, thus leading to a destructive interference part. It is note worthy that such a destructive

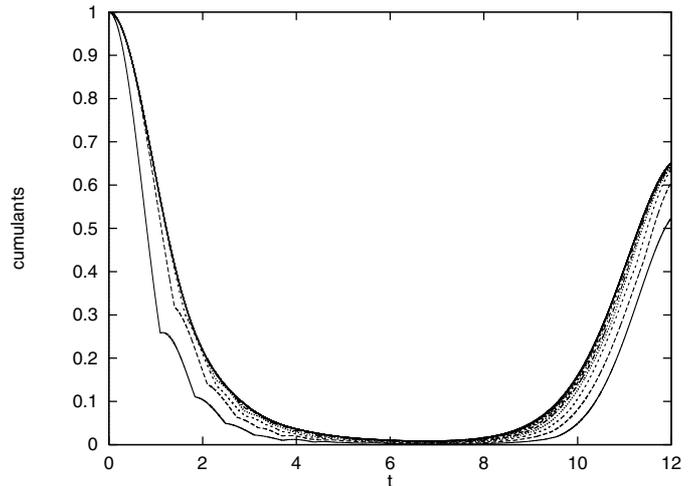


Fig. 12. Quantum walk algorithm. Time evolution of the 22 cumulants when one node per column is considered, for $n = 10$. Note that most of the quantities apparently collapse when making the plot because of the small difference in the probabilities given the big size of the graph.

interference indeed strictly follows a step-by-step reverse majorization of probabilities (satisfying the inequalities given in Eq. (2) for the case of reverse majorization). Furthermore, we see combining Figures 10 and 11 that the growing of the probability of success is linked to a step-by-step majorization. Physically, this is the part in which the algorithm constructively interferes into the OUT node once the wave packet is approximately in the right hand side binary tree. We see that this constructive interference follows a majorization arrow, thus verifying step-by-step the inequalities given in equation (2). Actually, the observed majorization cycle reminds us the one already found in the quantum algorithm of Section 3, but in this case we have numerically checked that the present cycle does not seem to follow the rules of natural majorization. Complementarily, we have also observed that the probability amplitudes follow the interesting rule that those belonging to even columns are real and those belonging to odd columns are imaginary.

The deterministic search by quantum random walk heavily exploits the column structure of the problem. The register works on a superposition of columns, that is of states belonging to the same column with equal weight. It is then natural to ask whether a step-by-step majorization cycle operates also at the level of columns. The idea behind this analysis corresponds to accept that the final measurement will filter each columns as a whole. The result of the measurement would correspond to determining a particular column. The subtle point here is to find to what extend the success of finding the OUT state is related to the column structure of the algorithm. We have numerically considered the column amplitudes for $n = 4$ and $n = 10$ with a total of 9 and 21 cumulants to be calculated respectively from the sorted probabilities at each time step of being *at each column* of the graph. In Figures 13 and 14 we plot the result, which shows that *there*

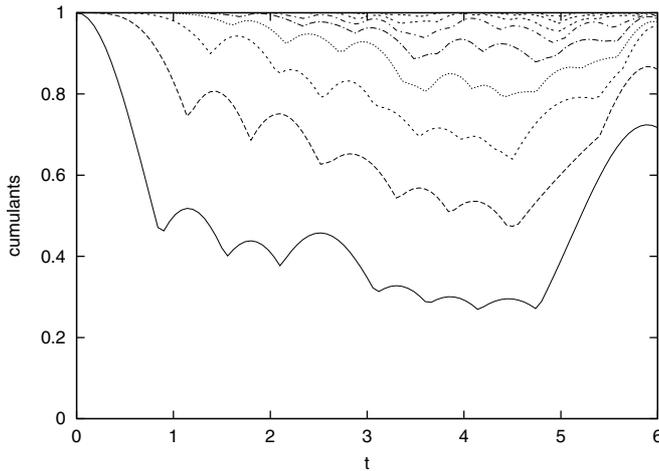


Fig. 13. Quantum walk algorithm. Time evolution of the nine cumulants when the column measurement is considered, for $n = 4$.

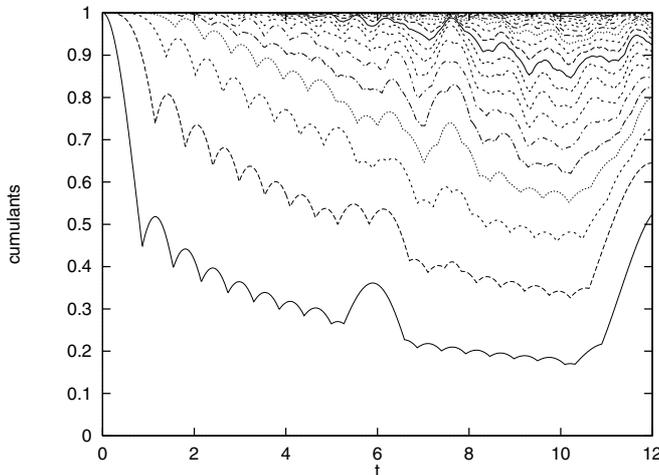


Fig. 14. Quantum walk algorithm. Time evolution of the 21 cumulants when the column measurement is considered, for $n = 10$.

does not exist a majorization cycle when the final measurement is made on columns.

The conclusion is that deterministic quantum walks cleverly exploit the column subspace structure of the problem to achieve step-by-step majorization on the individual states.

7 Conclusion: A majorization principle

We can now collect all the results found in the analysis of majorization in the quantum algorithms we have studied so far and synthesize an emerging principle underlying all of them. There are a total of nine empirical observations about step-by-step majorization:

- presence of step-by-step majorization in Grover’s quantum searching algorithm [16];
- presence of a natural step-by-step majorization cycle in quantum phase estimation algorithms [17];

- presence of natural step-by-step majorization cycle in the quantum algorithm for finding hidden affine functions (Sect. 3.3);
- absence of step-by-step majorization in an optimal, yet non-efficient, quantum algorithm for solving the parity problem (Sect. 4.3);
- absence of step-by-step majorization in naive global quantum adiabatic searching algorithms (Sect. 5.1);
- presence of step-by-step majorization in appropriate local quantum adiabatic searching algorithms (Sect. 5.2);
- emergence of step-by-step majorization for a slow enough evolution rate in quantum adiabatic searching algorithms (Sect. 5.3);
- absence as step-by-step majorization in an adiabatic quantum algorithm solving the 2-SAT “ring of agrees” problem (Sect. 5.4);
- presence of step-by-step majorization cycle in a deterministic quantum walk on a graph solving a classically hard problem (Sect. 6.3).

Note that our results concerning the analysis of adiabatic quantum algorithms can have an alternative valid interpretation: according to Figures 2 and 7, we see that the part of the processes which does not obey step-by-step majorization only occurs when the probability of success has almost achieved its highest value. We are then led to the consideration that absence of majorization only appears once the algorithms have already constructed the right solution, having already done their job. A redefinition of the algorithm by stopping the process once the probability of the winner is maximum would lead us to affirm that step-by-step majorization is naturally present in quantum algorithms by adiabatic evolution according to the evidence presented here. This new interpretation does not alter our final result.

Adiabatic algorithmic processes do lead as well to a reverse majorization of the probability distribution in order to efficiently prepare the initial quantum state of the computation. This can be efficiently performed by a set of Hadamard gates (which produce step-by-step natural reverse majorization). Nevertheless, this remark does not only hold for the adiabatic paradigm. The usual formulation of Grover’s algorithm in terms of quantum gates needs as well of a preparation of the initial quantum state which can be carried out exactly in the same way. Similarly, all algorithms accommodate to a reverse majorization-majorization cycle. We could argue that the initial step-by-step reverse majorization procedure at the beginning of the quantum algorithms is somehow trivial, as it only involves the application of (for instance) a set of Hadamard gates (with the exception of the quantum walk algorithm, in which the initial step-by-step reverse majorization is by no means trivial in the sense we state here as it is carried by the structure of the graph). The fact that the quantum evolution accommodates to a step-by-step reverse majorization-majorization cycle is reminiscent of the reversibility of these quantum algorithms.

All the results found so far suggest that a step-by-step reverse majorization-majorization cycle seems to be a

necessary condition for efficiency in quantum computational processes, although *not sufficient*. This can be promoted to a principle.

Majorization principle. *Optimal quantum algorithms must follow a majorization cycle.*

This principle fits well with all the observed results given so far. Note that those processes which are not optimal do not necessarily follow the majorization cycle pattern: the case of the optimal algorithm solving the parity problem does not, while the extremely slow and inefficient but majorizing adiabatic processes do. Step-by-step majorization may be viewed as a strong irreversibility condition for success probability necessary for optimal quantum algorithms.

All our results are also consistent with a stronger statement, namely that both step-by-step majorization and large entanglement complement each other and are needed for exponential speed-up. Entanglement brings the genuine quantum mechanical tool which has to be used in an optimal way, that is verifying step-by-step majorization.

We acknowledge financial support from the projects MCYT FPA2001-3598, GC2001SGR-00065, IST-1999-11053, PB98-0685 and BFM2000-1320-C02-01. One of us (R.O.) wishes to thank A. Childs, E. Deotto, E. Farhi, J. Goldstone, S. Gutmann, A. Landahl and R. Sharma for insightful comments and stimulating discussions.

References

1. L.K. Grover, Phys. Rev. Lett. **78**, 325 (1997)
2. P.W. Shor, Proc. 35th IEEE (IEEE press, Los Alamitos CA, 1994), p. 352; [quant-ph/9508027](#)
3. E. Farhi, J. Goldstone, S. Gutmann, M. Sipser, [quant-ph/0001106](#)
4. A.M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, D. Spielman, [quant-ph/0209131](#)
5. J. Ahn, T.C. Weinacht, P.H. Bucksbaum, Science **287**, 463 (2000)
6. P. Knight, Science **287**, 441 (2000)
7. S. Lloyd, [quant-ph/9903057](#)
8. R. Jozsa, N. Linden, [quant-ph/0201143](#)
9. A. Galindo, M.A. Martín-Delgado, Rev. Mod. Phys. **74**, 347 (2002); [quant-ph/0112105](#)
10. G. Vidal, [quant-ph/0301063](#)
11. R.F. Muirhead, Proc. Edinburg Math. Soc. **21**, 144 (1903)
12. G.H. Hardy, J.E. Littlewood, G. Pólya, *Inequalities* (Cambridge University Press, 1978)
13. A.W. Marshall, I. Olkin, *Inequalities: Theory of Majorization and its Applications* (Acad. Press Inc., 1979)
14. R. Bathia, *Matrix Analysis*, Graduate Texts in Mathematics (Springer-Verlag, 1996), Vol. 169
15. M.A. Nielsen, G. Vidal, Quant. Inform. Comput. **1**, 76 (2001)
16. J.I. Latorre, M.A. Martín-Delgado, Phys. Rev. A **66**, 022305 (2002); [quant-ph/0111146](#)
17. R. Orús, J.I. Latorre, M.A. Martín-Delgado, Quant. Inform. Proc. **4**, 283 (2003); [quant-ph/0206134](#)
18. E. Bernstein, U. Vazirani, Quant. Compl. Theor. SIAM J. Comp. **26**(5), 1411 (1997)
19. D. Deutsch, Proc. R. Soc. Lond. A **400**, 97 (1985)
20. R. Cleve, A. Ekert, C. Macchiavello, M. Mosca, Proc. R. Soc. Lond. A **454**, 339 (1998)
21. M. Mosca, Ph.D. thesis, 1999
22. M.A. Nielsen, I. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000)
23. D. Coppersmith, IBM Research Report Report 19642, 1994; [quant-ph/0201067](#)
24. E. Farhi, J. Goldstone, S. Gutmann, M. Sipser, Phys. Rev. Lett. **81**, 5442 (1998); [quant-ph/9802045](#)
25. R. Beals, H. Buhrman, R. Cleve, M. Mosca, R. de Wolf, Proc. of the 99th Annual Symposium on Foundations of Computer Science (FOCS'98) (IEEE press, Los Alamitos CA, 1998), pp. 352–361; [quant-ph/9802049](#)
26. S. Das, R. Kobes, G. Kunstatter, [quant-ph/0204044](#)
27. J. Roland, N.J. Cerf, Phys. Rev. A **65**, 042308 (2002); [quant-ph/0107015](#)
28. W. van Dam, M. Mosca, U. Vazirani, Proceedings of the 42nd Annual Symposium of Computer Science (IEEE press, 2001), pp. 279–287; [quant-ph/0206003](#)
29. J. Roland, N.J. Cerf, [quant-ph/0302138](#)
30. S. A. Cook, Proc. 3rd Ann. ACM Symp. on Theory of Computing (Association for Computing Machinery, New York, 1971), pp. 151–158
31. Y. Aharonov, L. Davidovich, N. Zagury, Phys. Rev. A **48**, 1687 (1993)
32. D. Aharonov, A. Ambainis, J. Kempe, U. Vazirani, Proc. of the 33rd ACM Symposium on the Theory of Computing (ACM Press, New York, 2001), p. 50
33. A. Ambainis, E. Bach, A. Nayak, A. Vishwanath, J. Watrous, Proc. 33rd Symposium on the Theory of Computing (ACM Press, New York, 2001), p. 37
34. E. Farhi, S. Gutmann, Phys. Rev. A **58**, 915 (1998); [quant-ph/9706062](#)
35. A.M. Childs, E. Farhi, S. Gutmann, Quant. Inform. Proc. **1**, 35 (2002); [quant-ph/0103020](#)
36. J. Kempe, [quant-ph/0205083](#)
37. N. Shenvi, J. Kempe, K. Birgitta Whaley, [quant-ph/0210064](#)