Adiabatic quantum search algorithm for structured problems

Jérémie Roland and Nicolas J. Cerf
Quantum Information and Communication, CP 165/59, Ecole Polytechnique, Université Libre de Bruxelles, 1050 Brussels, Belgium

(Received 4 April 2003; published 15 December 2003)

The study of quantum computation has been motivated by the hope of finding efficient quantum algorithms for solving classically hard problems. In this context, quantum algorithms by local adiabatic evolution have been shown to solve an unstructured search problem with a quadratic speedup over a classical search, just as Grover's algorithm. In this paper, we study how the structure of the search problem may be exploited to further improve the efficiency of these quantum adiabatic algorithms. We show that by nesting a partial search over a reduced set of variables into a global search, it is possible to devise quantum adiabatic algorithms with a complexity that, although still exponential, grows with a reduced order in the problem size.

DOI: 10.1103/PhysRevA.68.062312 PACS number(s): 03.67.Lx, 89.70.+c

I. INTRODUCTION

Grover's quantum algorithm solves an unstructured search problem in a time of order $\sqrt{N}$, where N is the dimension of the search space, which corresponds to a quadratic speedup over a classical search [1]. This algorithm is proved to be optimal in the case of unstructured search problems [2]. Naturally, it can also be used to solve a structured search problem with a quadratic speedup over a naive classical search that would exhaustively check every possible solution. However, exploiting the structure of the problem is well known to lead to better classical search algorithms. It is therefore tempting to imagine that better quantum search algorithms may be devised similarly by exploiting the problem structure. Following this, Cerf, Grover, and Williams showed that this could be done by partitioning the unknown variables into two (or more) sets and nesting a quantum search over one set into another search over two (or more) sets, yielding an average complexity of order $\sqrt{N^{\alpha}}$, with $\alpha<1$ [3].

While this algorithm, as well as Grover's original algorithm, stay within the standard paradigm of quantum computation based on quantum circuits, a different type of quantum algorithm based on adiabatic evolution has been introduced lately [4]. In particular, a quantum adiabatic analogue of Grover's search algorithm has been independently developed in Refs. [5] and [6], which works for unstructured search problems. The use of quantum adiabatic algorithms has also been analyzed for solving structured problems such as k-satisfiability (k-SAT), but in such a way that until now only a numerical study has been possible [7]. Recently, the study of quantum adiabatic algorithms progressed even further after Aharonov and Ta-Shma demonstrated that any quantum state that may be efficiently generated in the quantum-circuit model can also be efficiently generated by an adiabatic quantum state generation algorithm [8]. This result could hopefully lead to the proof of the universality of algorithms by quantum adiabatic evolution and thus provides a strong incentive in the search for further quantum adiabatic algorithms.

The purpose of this paper is to bring the ideas of nested quantum search and quantum adiabatic computation together, in order to devise a quantum adiabatic algorithm adapted to structured problems. More specifically, we will show that an adiabatic search over a subset of the variables can be used to build a better initial Hamiltonian for the global adiabatic search. With this adiabatic algorithm, we recover the same complexity as for the nested circuit-based algorithm of Ref. [3], although we will see that it is slightly more general in that it does not require the exact number of solutions (and partial solutions) to be known *a priori*.

II. ADIABATIC THEOREM

Let us briefly recall the adiabatic theorem and how it may be used to design quantum algorithms by adiabatic evolution.

We know that if a quantum system is prepared in the ground state of the time-independent Hamiltonian driving its evolution, it remains in this state. The adiabatic theorem states that, if this Hamiltonian becomes time dependent, the system will still stay close to its instantaneous ground state as long as the variation is *slow enough*.

More specifically, if $|E_0;t\rangle$ and $|E_1;t\rangle$ are the ground and first excited states of the Hamiltonian $H(t)$, with energies $E_0(t)$ and $E_1(t)$, we define the minimum gap between these eigenvalues as

$$g_{\min}=\min_{0\leqslant t\leqslant T}[E_1(t)-E_0(t)] \tag{1}$$

and the maximum value of the matrix element of dH/dt between the eigenstates as

$$D_{\max}=\max_{0\leqslant t\leqslant T}\left|\left\langle\frac{dH}{dt}\right\rangle_{1,0}\right| \tag{2}$$

with $\langle dH/dt\rangle_{1,0}=\langle E_1;t|dH/dt|E_0;t\rangle$. The adiabatic theorem states that, if we prepare the system at time $t=0$ in its ground state $|E_0;0\rangle$ and let it evolve under the Hamiltonian $H(t)$, then

$$|\langle E_0;T|\psi(T)\rangle|^2\geqslant 1-\varepsilon^2 \tag{3}$$

provided that

$$\frac{D_{\max}}{g_{\min}^2} \leqslant \varepsilon, \tag{4}$$

where $\varepsilon \ll 1$.

Now, we may apply to the system a Hamiltonian for which the ground state encodes the unknown solution of a problem. According to the adiabatic theorem, we know that we may get the system very close to this solution state by preparing it in the (known) ground state of another Hamiltonian, and then by progressively changing it to the Hamiltonian of our problem. This simple idea is central to the quantum algorithms by adiabatic evolution [4,5].

III. QUANTUM SEARCH BY LOCAL ADIABATIC EVOLUTION

As exposed in [6], this principle may be used to perform a quantum search. Suppose that among N states, we have to find the M-times degenerate ground state of a Hamiltonian

$$H_f = I - \sum_{m \in \mathcal{M}} |m\rangle\langle m|, \tag{5}$$

where $\mathcal{M}$ is the ensemble of solutions (of size M). We initially prepare the system in an equal superposition of all could-be solutions:

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{i \in \mathcal{N}} |i\rangle. \tag{6}$$

This superposition is the ground state of the following Hamiltonian:

$$H_i = I - |s\rangle\langle s|. \tag{7}$$

We now apply H_i to the system and switch adiabatically to H_f. If we perform an adiabatic evolution

$$H(t) = [1 - s(t)]H_i + s(t)H_f, \tag{8}$$

where $s(t)$ is a (carefully chosen) monotonic function with $s(0)=0$ and $s(T)=1$, we will finally obtain a state close to a ground state of H_f:

$$|\psi_f\rangle \approx \frac{1}{\sqrt{M}} \sum_{m \in \mathcal{M}} |m\rangle \tag{9}$$

as long as

$$T = O\left(\sqrt{\frac{N}{M}}\right). \tag{10}$$

This algorithm is referred to as *local* because $s(t)$ is chosen such that the adiabatic theorem is obeyed locally, at each time (see [6] for details).

Note that if there is more than one solution ($M>1$) each solution corresponds to a ground state of H_f and the system gets to the uniform superposition of all of these states (9) because the whole problem is symmetric under permutation of the solution states $|m\rangle$. If the number of solutions M is unknown, we may use in Eq. (10) an arbitrary value M' of the order of M which will affect only the error probability of the computation by a factor of M'/M (if $M'=1$ is chosen, then the error probability can only be lower than with $M'=M$, but the computation time will be longer). This is a major difference to Grover's conventional algorithm, where the computation has to be run several times when the number of solutions is unknown, and will be helpful in our structured search.

IV. STRUCTURED PROBLEMS

In this article, we consider a class of problems where one has to find an assignment for a set of variables. For each additional variable considered, new constraints appear and reduce the set of satisfying assignments. This corresponds to most problems encountered in practice (k-SAT, graph coloring, planning, combinatorial optimization, etc.).

For a set of n_A variables denoted as A, there is a corresponding set of constraints C_A. We may define a function f_A that tells if an assignment of the variables in A satisfies the constraints in C_A:

$$\begin{aligned} f_A &: (\mathbb{Z}_d)^{n_A} \to \{0,1\} \\ &: x \to \begin{cases} 0 & \text{if } x \text{ does not satisfy } C_A, \\ 1 & \text{if } x \text{ satisfies } C_A, \end{cases} \end{aligned} \tag{11}$$

where d is the number of possible assignments for each variable ($d=2$ for bits). As quantum gates have to be reversible, the quantum equivalent of this function will be an oracle:

$$O_A : \mathcal{H}_{N_A} \otimes \mathcal{H}_2 \to \mathcal{H}_{N_A} \otimes \mathcal{H}_2 : |x\rangle \otimes |y\rangle \to |x\rangle \otimes |y \oplus f_A(x)\rangle, \tag{12}$$

where $N_A = d^{n_A}$. It is shown in Ref. [9] that this oracle is closely related to a Hamiltonian whose ground states, of energy 0, are the basis states encoding a satisfying assignment and whose excited states, of energy 1, are all other basis states:

$$H_A|x\rangle = \begin{cases} |x\rangle & \text{if } x \text{ does not satisfy } C_A, \\ 0 & \text{if } x \text{ satisfies } C_A, \end{cases} \tag{13}$$

or

$$H_A = I_A - \sum_{m_A \in \mathcal{M}_A} |m_A\rangle\langle m_A|, \tag{14}$$

where $\mathcal{M}_A$ is the set of satisfying assignments for the variables in A. It is possible to efficiently simulate the time evolution according to this Hamiltonian, that is, the unitary operator e^{-iHt} can be well approximated using a sequence of one- and two-qubit gates and two oracle calls (see [9] for details).

Now suppose we consider a larger set of variables $n_{AB} = n_A + n_B$ that have to satisfy a set of constraints $C_{AB} \supset C_A$. To discriminate between assignments satisfying C_{AB} or not,

we will use an oracle O_{AB} or a corresponding Hamiltonian H_{AB} defined as in Eqs. (12) and (13). The basic idea of our structured search will be to find the solutions of C_{AB} by first building the assignments of the n_A primary variables satisfying C_A, then by completing them with all possible assignments of the n_B secondary variables, and finally by searching among these could-be solutions the global satisfying assignments.

V. STRUCTURED SEARCH BY NESTED ADIABATIC EVOLUTION

This problem is of the same type as the one considered in [3], for which the technique of nesting was introduced in the context of the traditional implementation of Grover's algorithm on a conventional quantum circuit. Here, we apply this technique to the adiabatic quantum search algorithm.

Suppose we divide the variables of our problem into two subsets A (n_A elements) and B (n_B elements). First, we will perform a search on the variables in A using the Hamiltonian H_A that encodes the constraints in C_A:

$$H_A = I_A - \sum_{m_A \in \mathcal{M}_A} |m_A\rangle\langle m_A|. \tag{15}$$

Then we will use the Hamiltonian H_{AB} acting on all variables in $A \cup B$ and encoding the whole set of constraints C_{AB}

$$H_{AB} = I_{AB} - \sum_{(m_A,m_B) \in \mathcal{M}_{AB}} |m_A\rangle\langle m_A| \otimes |m_B\rangle\langle m_B| \tag{16}$$

to construct a superposition of the solutions of the full problem $\mathcal{M}_{AB}$. A final measurement of the quantum register then gives one of the global solutions at random.

A. Adiabatic search on the primary variables

The preliminary search on the variables in A is a simple unstructured search as explained in Sec. III. As there are n_A variables in A, the corresponding Hilbert space is of dimension $N_A = d^{n_A}$. Let M_A be the number of solutions in $\mathcal{M}_A$. Performing an adiabatic quantum search, we may thus transform the initial state

$$|s_A\rangle = \frac{1}{\sqrt{N_A}} \sum_{i \in \mathcal{N}_A} |i\rangle_A \tag{17}$$

into a state close to the uniform superposition of all solutions in $\mathcal{M}_A$,

$$|\psi_{m_A}\rangle = \frac{1}{\sqrt{M_A}} \sum_{m_A \in \mathcal{M}_A} |m_A\rangle, \tag{18}$$

in a time of order

$$T_A = O\left(\sqrt{\frac{N_A}{M_A}}\right). \tag{19}$$

Let us point out that, here and throughout the rest of the article, it seems that the number of solutions M_A (and later M_{B/m_A} and M_A^S) must be known to derive the minimal time T_A (and later T_B and T_C) needed to perform the computation with a bounded error probability. Actually, as already explained in the case of the unstructured search at the end of Sec. III, an approximate value M' of the order of the actual M is sufficient as it will affect the error probability only by a factor of M'/M. In real problems, this issue may thus be addressed by using approximate methods to evaluate the number of solutions (such as Eq. (57) of Sec. VII for k-SAT).

B. Adiabatic search on the secondary variables

We will now perform a preliminary search in the Hilbert space of dimension $N_B = d^{n_B}$ of the secondary variables in B by extending the partial solutions $|m_A\rangle$. We prepare the variables in B in a state that is the uniform superposition

$$|s_B\rangle = \frac{1}{\sqrt{N_B}} \sum_{j \in \mathcal{N}_B} |j\rangle_B. \tag{20}$$

Globally, the system is thus in the superposition

$$|\psi_0\rangle_{AB} = |\psi_{m_A}\rangle \otimes |s_B\rangle = \frac{1}{\sqrt{M_A N_B}} \sum_{\substack{m_A \in \mathcal{M}_A \\ j \in \mathcal{N}_B}} |m_A\rangle \otimes |j\rangle_B, \tag{21}$$

where some terms correspond to a global solution of the problem [$(m_A, j) \in \mathcal{M}_{AB}$ satisfying all constraints in C_{AB}] and the others to a partial solution only [$m_A \in \mathcal{M}_A$ satisfies C_A but $(m_A, j) \notin \mathcal{M}_{AB}$ does not satisfy C_{AB}]. We now divide the set $\mathcal{M}_A$ of solutions of C_A into two subsets: $\mathcal{M}_A^S$ will be the set of m_A's for which there exists at least one solution (m_A, m_B) of C_{AB} and $\mathcal{M}_A^{NS}$ the set of m_A's for which there is no such solution,

$$\mathcal{M}_A^S = \{m_A \in \mathcal{M}_A | \exists m_B, (m_A, m_B) \in \mathcal{M}_{AB}\}, \tag{22}$$

$$\mathcal{M}_A^{NS} = \{m_A \in \mathcal{M}_A | \forall j, (m_A, j) \notin \mathcal{M}_{AB}\}. \tag{23}$$

Of course, we thus have $\mathcal{M}_A = \mathcal{M}_A^S \cup \mathcal{M}_A^{NS}$. We may now rewrite our initial state (21) as

$$|\psi_0\rangle_{AB} = \frac{1}{\sqrt{M_A N_B}} \sum_{\substack{m_A \in \mathcal{M}_A^{NS} \\ j \in \mathcal{N}_B}} |m_A\rangle \otimes |j\rangle_B + \frac{1}{\sqrt{M_A N_B}} \sum_{\substack{m_A \in \mathcal{M}_A^{S} \\ j \in \mathcal{N}_B}} |m_A\rangle \otimes |j\rangle_B. \tag{24}$$

In the first part of this expression, no term corresponds to a solution of the full problem, whereas in the second part, some terms do and others do not. The goal of this stage of the computation will be to increase the amplitude of the so-

lution terms in this last part. To achieve this, we perform an adiabatic evolution using as initial Hamiltonian

$$H_i = I_A \otimes (I_B - |s_B\rangle\langle s_B|), \tag{25}$$

that has $|\psi_0\rangle_{AB}$ as a ground state. The final Hamiltonian will be

$$H_f = H_{AB} - H_A \otimes I_B . \tag{26}$$

We see that these Hamiltonians share the following properties.

(1) They do not induce evolution of states $|i\rangle_A \otimes |s_B\rangle$ corresponding to assignments i of $\mathcal{N}_A$ that do not satisfy C_A: $H_{i,f}|i\rangle_A \otimes |s_B\rangle = 0 \quad \forall \quad i \notin \mathcal{M}_A$.

(2) They do not couple states corresponding to different m_A's: $_B\langle j| \otimes \langle m_A|H_{i,f}|m'_A\rangle \otimes |j'\rangle_B = 0, \quad \forall \; m_A \neq m'_A \in \mathcal{M}_A$, $\forall \; j,j' \in \mathcal{N}_B$.

It follows that the instantaneous Hamiltonian of the adiabatic evolution $H(t)$ satisfies the same properties. Keeping this in mind, it may easily be shown that the effect of the adiabatic evolution will be to perform independent adiabatic searches for each $m_A \in \mathcal{M}_A$. More precisely, each term in $|\psi_0\rangle_{AB}$

$$\frac{1}{\sqrt{N_B}} \sum_{j \in \mathcal{N}_B} |m_A\rangle \otimes |j\rangle_B \tag{27}$$

will evolve to

$$\frac{1}{\sqrt{M_{B/m_A}}} \sum_{m_B \in \mathcal{M}_{B/m_A}} |m_A\rangle \otimes |m_B\rangle, \tag{28}$$

as long as

$$T_{m_A} = O\left(\sqrt{\frac{N_B}{M_{B/m_A}}} \right), \tag{29}$$

where $\mathcal{M}_{B/m_A}$ is the set of m_B's such that $(m_A, m_B) \in \mathcal{M}_{AB}$ and M_{B/m_A} is the number of these elements. For this condition to be satisfied for all m_A's simultaneously, we must take

$$T_B = \max_{m_A} T_{m_A} = O\left(\sqrt{\frac{N_B}{\min\limits_{m_A} M_{B/m_A}}} \right). \tag{30}$$

Here is the major advantage of this adiabatic algorithm compared to its circuit-based counterpart [3] where all M_A's had to be supposed equal to 1, as here it is sufficient that they are of the same order of magnitude to ensure an error probability of the same order for each term.

At the end of this second stage, we have thus constructed a state close to

$$|\psi_{AB}\rangle = \frac{1}{\sqrt{M_A N_B}} \sum_{\substack{m_A \in \mathcal{M}_A^{NS} \\ j \in \mathcal{N}_B}} |m_A\rangle \otimes |j\rangle_B + \frac{1}{\sqrt{M_A}} \sum_{m_A \in \mathcal{M}_A^S} e^{i\phi_{m_A}} |m_A\rangle \otimes \frac{1}{\sqrt{M_{B/m_A}}} \sum_{m_B \in \mathcal{M}_{B/m_A}} |m_B\rangle = \sqrt{\frac{M_A^{NS}}{M_A}} |\psi^{NS}\rangle + \sqrt{\frac{M_A^S}{M_A}} |\psi^S\rangle, \tag{31}$$

where the ϕ_{m_A}'s are phases appearing during the evolution,

$$|\psi^{NS}\rangle = \frac{1}{\sqrt{M_A^{NS} N_B}} \sum_{\substack{m_A \in \mathcal{M}_A^{NS} \\ j \in \mathcal{N}_B}} |m_A\rangle \otimes |j\rangle_B , \tag{32}$$

$$|\psi^S\rangle = \frac{1}{\sqrt{M_A^S}} \sum_{m_A \in \mathcal{M}_A^S} e^{i\phi_{m_A}} |m_A\rangle \otimes \frac{1}{\sqrt{M_{B/m_A}}} \sum_{m_B \in \mathcal{M}_{B/m_A}} |m_B\rangle, \tag{33}$$

and M_A^{NS} (M_A^S) is the number of elements in set $\mathcal{M}_A^{NS}$ ($\mathcal{M}_A^S$).

C. Global adiabatic search

The stages A and B define a unitary evolution U that applies the initial state $|s_A\rangle \otimes |s_B\rangle$ onto $|\psi_{AB}\rangle$:

$$U|s_A\rangle \otimes |s_B\rangle \approx |\psi_{AB}\rangle \tag{34}$$

$$= \sqrt{\frac{M_A^{NS}}{M_A}} |\psi^{NS}\rangle + \sqrt{\frac{M_A^S}{M_A}} |\psi^S\rangle. \tag{35}$$

In this state, we now need to decrease the amplitude of the first term, corresponding to partial solutions only, and increase the amplitude of the second term, corresponding to global solutions. This could be realized efficiently by performing an adiabatic search using as initial Hamiltonian:

$$H_i = I_{AB} - |\psi_{AB}\rangle\langle\psi_{AB}| \tag{36}$$

$$\approx U(I_{AB} - |s_A\rangle\langle s_A| \otimes |s_B\rangle\langle s_B|)U^\dagger \tag{37}$$

$$\approx U H_0 U^\dagger, \tag{38}$$

where $H_0 = I_{AB} - |s_A\rangle\langle s_A| \otimes |s_B\rangle\langle s_B|$, and as final Hamiltonian

$$H_f = H_{AB} = I_{AB} - \sum_{(m_A, m_B) \in \mathcal{M}_{AB}} |m_A\rangle\langle m_A| \otimes |m_B\rangle\langle m_B| \tag{39}$$

during a time

$$T_C = 0\left(\sqrt{\frac{M_A}{M_A^S}} \right). \tag{40}$$

Unfortunately, we do not have access to H_i, so that the interpolating Hamiltonian $H(s)=(1-s)H_i+sH_f$ cannot be applied directly. However, we will see in Sec. VI that the basic steps of the quantum-circuit implementation of this adiabatic algorithm require only the application of H_i during a particular time t, that is,

$$e^{-iH_i t} \approx e^{-iUH_0U^\dagger t} = Ue^{-iH_0 t}U^\dagger. \tag{41}$$

Hence, each application of H_i during a time t will be equivalent to sequentially applying $U^\dagger$, $e^{-iH_0 t}$, and U, which means performing the adiabatic evolution U (stages A and B) backward, then applying H_0 for a time t, and finally rerun U forward (stages A and B).

In Sec. VI, we will see that, when discretizing the evolution, we must take a number of steps r_C of order T_C. We may now evaluate the complexity of this algorithm. As it consists of r_C steps, each involving two applications of U or $U^\dagger$, that last a time of order T_A+T_B, the algorithm finally takes a time of order

$$\begin{aligned} T &= (T_A+T_B)r_C \qquad (42)\\ &= O\left(\left(\sqrt{\frac{N_A}{M_A}} + \sqrt{\frac{N_B}{\min\limits_{m_A} M_{B/m_A}}} \right) \sqrt{\frac{M_A}{M_A^S}} \right)\\ &= O\left(\sqrt{\frac{N_A}{M_A^S}} + \sqrt{\frac{M_A N_B}{M_A^S \min\limits_{m_A} M_{B/m_A}}} \right). \qquad (43) \end{aligned}$$

Let us notice that, with the same hypothesis as in Ref. [3], namely,

$$M_{B/m_A} = 1 \quad \forall \quad m_A, \tag{44}$$

$M_A^S = M_{AB}$, and the computation time is

$$T = O\left(\frac{\sqrt{N_A}+\sqrt{M_A N_B}}{\sqrt{M_{AB}}} \right), \tag{45}$$

so that the complexity is the same as that of the equivalent circuit-based algorithm described in Ref. [3]. A more detailed analysis of this complexity will be performed in Sec. VII.

VI. DISCRETIZING THE ADIABATIC EVOLUTION

A. General method

The implementation of a *global* adiabatic evolution algorithm on a discrete quantum circuit was initially shown in [4], further studied in [5], and extended to the case of a *local* adiabatic evolution algorithm in [9]. Let us recall that we use the term ''global'' when the adiabatic condition is imposed globally and the evolution interpolates linearly between the initial and the final Hamiltonians, whereas ''local'' means that the evolution is optimized at each time, using a local version of the adiabatic condition, which is the case here. We now quickly review the discretization of this last method, which uses two successive approximations.

The first approximation consists in cutting the evolution time T into r intervals $\Delta T = T/r$ and replacing the continuously varying Hamiltonian $H(t)$ by a Hamiltonian $H'(t)$ that is constant during each interval ΔT and varies at times $t_j = j\Delta T$ only:

$$H'(t) = H(t_j) \quad \text{if} \quad t_{j-1} \leqslant t \leqslant t_j. \tag{46}$$

It is shown in [9] that, for $H(s)=(1-s)H_i+sH_f$ with $s=s(t)$, this approximation introduces a global error on the corresponding evolution such that

$$|||U(T)-U'(T)|||_2 \leqslant \sqrt{2\frac{T}{r}|||H_i-H_f\vdots\cdot|_2}, \tag{47}$$

where $|||A|||_2 = \max_{|||x\rangle||=1} |||A|x\rangle||$ is the operator norm of A. Our algorithm now requires r steps of the form

$$U_j' = e^{-iH(t_j)\Delta T} = e^{-i(1-s_j)H_i\Delta T - is_jH_f\Delta T}, \tag{48}$$

where $s_j = s(t_j)$. As we are able to apply H_i and H_f separately but not necessarily a simultaneous combination of them, we will approximate U_j' by

$$U_j'' = e^{-i(1-s_j)H_i\Delta T}e^{-is_jH_f\Delta T}. \tag{49}$$

This will result in an error

$$|||U'(T) - \prod_j U_j''|||_2 \in O\left(\frac{T^2}{r}|||[H_i,H_f]|||_2 \right) \tag{50}$$

(see [9] for details).

B. Application to a structured quantum search

We now consider the case of a structured quantum search. We could apply the discretization procedure to all three stages (A,B,C) of our algorithm in order to implement it on a quantum circuit, but we will concentrate on stage C, which is the only one that requires discretization. Nonetheless, it is easy to show that stage $A(B)$ would require a number of steps r_A (r_B) of the same order as the computation time T_A (T_B).

For the final stage, the global adiabatic search, the Hamiltonians H_i and H_f are defined in Eqs. (36)–(39). Evaluating the errors introduced by the approximations, we find

$$|||H_i-H_f|||_2 < 1, \tag{51}$$

$$|||[H_i,H_f]|||_2 < \sqrt{\frac{M_A^S}{M_A}}, \tag{52}$$

and, as $T_C = 0(\sqrt{M_A/M_A^S})$,

$$\| \| U(T) - U'(T) \| \|_2 \in O\left(\sqrt{\frac{\sqrt{M_A/M_A^S}}{r_C}} \right), \tag{53}$$

$$\| \| U'(T) - \prod_j U_j'' \| \|_2 \in O\left(\frac{\sqrt{M_A/M_A^S}}{r_C} \right). \tag{54}$$

Therefore, as announced in Sec. V, we have to cut our evolution into a number of steps $r_C = O(\sqrt{M_A/M_A^S})$ of the same order as T_C. Each step j will take the form

$$U_j'' = e^{-i(1-s_j)H_i \Delta T} e^{-i s_j H_f \Delta T} \tag{55}$$

$$\approx U e^{-i(1-s_j)H_0 \Delta T} U^\dagger e^{-i s_j H_f \Delta T}, \tag{56}$$

where the applications of Hamiltonians H_0 during a time $(1-s_j)\Delta T$ and H_f during a time $s_j \Delta T$ may be realized by the procedure described in [9].

VII. COMPLEXITY ANALYSIS

To estimate the efficiency of this algorithm, we will follow the same development as in [3]: as we have seen in Sec. V, under the assumption (44) that we will consider here, the complexity of this adiabatic algorithm has exactly the same form as its circuit-based counterpart.

First of all let us define a few concepts (for details here and throughout this section, we refer the reader to Ref. [3]). The structured search problem is to find an assignment of $n_{AB} = n_A + n_B$ variables among d possibilities and satisfying e constraints, each involving at most k of these variables. We define as a *ground instance* an assignment of all the variables involved in a particular constraint. A ground instance will be said to be *no good* if it violates the constraint. Let ξ be the number of those no-good ground instances.

Empirical studies show that the difficulty of solving a structured problem essentially depends on four parameters: the number of variables n_{AB}, the number of possible assignment per variable d, the number of variables per constraint k, and the number of no-good ground instances ξ. Intuitively, we understand that if ξ is small, there are many assignments satisfying the constraints so the problem is easy to solve. On the contrary, if ξ is large, the problem is overconstrained and it is easy to show that there is no solution. More precisely, it may be shown that for fixed n_{AB} and d, the average difficulty may be evaluated by the parameter $\beta = \xi / n_{AB}$. The hard problems will be concentrated around a critical value β_c.

Let us now estimate the complexity (45). Let $p(n)$ be the probability that a randomly generated assignment of the n first variables satisfies all the constraints involving these variables. We then have $M_A = p(n_A) d^{n_A}$ and $M_{AB} = p(n_{AB}) d^{n_{AB}}$ while it is shown in [3] that

$$p(n) \approx d^{-n_{AB}(\beta/\beta_c)(n/n_{AB})^k}. \tag{57}$$

Equation (45) becomes

$$T = O\left(\frac{\sqrt{d^{n_A}} + \sqrt{d^{n_{AB}[1-(\beta/\beta_c)(n_A/n_{AB})^k]}}}{\sqrt{d^{n_{AB}(1-\beta/\beta_c)}}} \right) \tag{58}$$

or, with $a = \sqrt{d^{n_{AB}}}$ and $x = n_A / n_{AB}$,

$$T = O\left(\frac{a^x + a^{1-(\beta/\beta_c)x^k}}{a^{1-\beta/\beta_c}} \right). \tag{59}$$

We now optimize x, the fraction of variables for which we perform a partial search, to minimize the computation time. We have to solve the equation

$$\frac{\beta}{\beta_c} k x^{k-1} = a^{(\beta/\beta_c)x^k + x - 1}, \tag{60}$$

which, for large a (that is, large n_{AB}) approximately reduces to

$$\frac{\beta}{\beta_c} x^k + x - 1 = 0. \tag{61}$$

The solution of this equation α $(0 \leqslant \alpha \leqslant 1)$ corresponds to the optimal partial search we may perform such that the complexity grows with the smallest power in d for $n_{AB} \to \infty$. This optimal computation time may then be written as

$$T = O\left(\frac{2a^\alpha}{a^{1-\beta/\beta_c}} \right) = O\left(\frac{\sqrt{d^{\alpha n_{AB}}}}{\sqrt{d^{n_{AB}(1-\beta/\beta_c)}}} \right). \tag{62}$$

Let us now consider the hardest problems for which $\beta \approx \beta_c$. For these problems, the complexity reads

$$T = O(\sqrt{d^{\alpha n_{AB}}}), \tag{63}$$

which we may immediately compare to the complexity of an unstructured quantum search $O(\sqrt{d^{n_{AB}}})$. The gain in the exponent α depends on k through Eq. (61). For instance, we find $\alpha = 0.62$ for $k=2$, $\alpha = 0.68$ for $k=3$, and $\alpha \to 1$ when $k \to \infty$.

As already pointed out, we recover exactly the same complexity as for the circuit-based structured search algorithm shown in [3], but with fewer hypotheses as, due to the particular form of the required running time for an adiabatic algorithm (10), the number of solutions derived from Eq. (57) must give only an order of magnitude, while it must be a good approximation for its circuit-based analogue. Moreover, as seen in Sec. V, the numbers of solutions M_{B/m_A} do not have to be equal for all m_A's, but only of the same order.

To compare these results with a classical algorithm, let us consider a specific problem, the satisfiability of Boolean formulas in conjunctive normal form, or k-SAT. For 3-SAT, which is known to be *NP* complete, some of the best classical algorithms have a worst-case running time that scales as $O(2^{0.4 n_{AB}})$ [10,11], while, as $\alpha = 0.68$ for $k=3$, our quantum adiabatic algorithm has a computation time of order $O(2^{0.34 n_{AB}})$, which is a slight improvement. Nonetheless, let us recall that there is a distinction between the worst-case complexity used for characterizing classical algorithms and the average-case complexity for hardest problems $(\beta = \beta_c)$ used for characterizing our quantum algorithm. However, let us also notice that this scaling could be further improved by using several levels of nesting, i.e., by replacing the prelimi-

nary search over the primary variables by another nested structured search (see the analysis of the circuit-based counterpart of this idea in the Appendix of [3]).

VIII. CONCLUSION

We have introduced a quantum search algorithm combining the approach based on local adiabatic evolution developed in [6] and the nesting technique introduced in [3]. It allows one to adiabatically solve structured search problems with an improved complexity over a naive adiabatic search that would not exploit the structure of the problem.

The basic idea is to perform a preliminary adiabatic search over a reduced number of variables of the problem in order to keep only a superposition of the assignments that respect the constraints of this partial problem, and then to complete these partial solutions by finding satisfying assignments for the remaining variables. We have seen that, to implement this algorithm, the global adiabatic evolution (stage *C*) has to be discretized, which makes it possible to *nest* the preliminary adiabatic search (stages *A* and *B*) into the global one. Each step of the discretized algorithm requires alternating partial adiabatic searches backward and forward with global search iteration steps.

A complexity analysis shows that the average computation time of this adiabatic algorithm, although still exponential, grows with a reduced exponent compared to quantum unstructured search algorithms to solve a problem such as 3-SAT.

ACKNOWLEDGMENTS

J.R. acknowledges support from the Belgian foundation FRIA. This work was funded in part by the Communauté Française de Belgique under Grant No. ARC 00/05-251, by the IUAP program of the Belgian government under Grant No. V-18, and by the EU under project RESQ (Grant No. IST-2001-35759).

[1] L.K. Grover, in *Proceedings of the 28th Annual Symposium on the Theory of Computing* (ACM Press, New York, 1996), pp. 212–219.

[2] C.H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, SIAM J. Comput. **26**, 1510 (1997).

[3] N.J. Cerf, L.K. Grover, and C.P. Williams, Phys. Rev. A **61**, 032303 (2000).

[4] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, e-print quant-ph/0001106.

[5] W. van Dam, M. Mosca, and U. Vazirani, in *Proceedings of the 42nd Annual Symposium on the Foundations of Computer Science* (IEEE Computer Society Press, New York, 2001), pp. 279–287.

[6] J. Roland and N.J. Cerf, Phys. Rev. A **65**, 042308 (2002).

[7] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, and A. Lundgren, Science **292**, 472 (2001).

[8] D. Aharonov and A. Ta-Shma, in *Proceedings of the 35th Annual ACM Symposium on Theory of Computing* (ACM Press, New York, 2003), pp. 20–29.

[9] J. Roland and N.J. Cerf, preceding paper, Phys. Rev. A **68**, 062311 (2003).

[10] U. Schöning, in *Proceedings of the 40th Annual Symposium on the Foundations of Computer Science* (IEEE Computer Society Press, New York, 1999), pp. 410–414.

[11] T. Hofmeister, U. Schöning, R. Schuler, and O. Watanabe, in *Proceedings of the 19th Annual Symposium on Theoretical Aspects of Computer Science*, edited by H. Alt and A. Ferreira, Lecture Notes in Computer Science Vol. 2285 (Springer, Berlin, 2002), pp. 192–202.