A quantum differentiation of \( k \)-SAT instances

B Tamir\(^1\) and G Ortiz\(^2,3\)

\(^1\) Israel Institute for Advanced Research, Rehovot, Israel
\(^2\) Department of Physics, Indiana University, Bloomington, IN 47405, USA
E-mail: canjlm@actcom.co.il and ortizg@indiana.edu

\textit{New Journal of Physics} \textbf{12} (2010) 075028 (10pp)
Received 24 March 2010
Published 28 July 2010
Online at \texttt{http://www.njp.org/}
doi:10.1088/1367-2630/12/7/075028

\textbf{Abstract.} We present a quantum adiabatic algorithm to differentiate between \( k \)-SAT instances, those with no solutions and those that have many solutions. The time complexity of the algorithm is a function of the energy gap between the subspace of all 0-eigenvectors (ground states) and the first excited states manifold, and scales polynomially with the number of resources. The idea of gaps between subspaces suggests a new tool to analyze time complexity in adiabatic quantum machines.

\textbf{Contents}

1. Introduction and motivations \hspace{1cm} 2
2. The quantum adiabatic machine \hspace{1cm} 3
   2.1. The state space \hspace{1cm} 3
   2.2. The problem Hamiltonian \hspace{1cm} 4
   2.3. The initial Hamiltonian \hspace{1cm} 5
   2.4. The quantum adiabatic algorithm \hspace{1cm} 5
3. The complexity of the algorithm: the evolution time \hspace{1cm} 6
4. Discussion \hspace{1cm} 9
References \hspace{1cm} 10

\(^3\) Author to whom any correspondence should be addressed.
1. Introduction and motivations

In [1], Farhi et al suggested an adiabatic quantum algorithm to solve certain search-type problems. Such kinds of algorithms are based on the well-known adiabatic theorem of quantum mechanics [2, 3]. The adiabatic quantum machine implementing the algorithm turns out to be the quantum parallel of the simulated annealing classical computer [4]. In [5, 6], it was shown that adiabatic quantum computation is polynomially equivalent to standard quantum computation (i.e. to the qubit circuit model [7]). Nonetheless, there are only a few practical examples of adiabatic quantum algorithms (see also [5]).

In an adiabatic quantum computer, one evolves from a simple and easy to construct initial vector towards a final vector using a time-dependent Hamiltonian. The final Hamiltonian is constructed so that its 0-eigenvector, its ground state, is a solution to the computational problem. If the initial vector is a 0-eigenvector of the initial Hamiltonian and the evolution is done slowly enough, then the final vector is supposed to be the 0-eigenvector of the final Hamiltonian. This is the content of the quantum adiabatic theorem.

Our Hamiltonian resembles the one used by Farhi et al [1]. It is a sum of quasi-local operators in terms of Pauli spin-1/2 matrices. This Hamiltonian is $k$-local for some small $k$. In its diagonal form, each of the $2^n$ assignments, where $n$ is the number of Boolean variables or spins, represents a penalty function for each $k$-SAT clause that the corresponding assignment does not satisfy and is 0 if this assignment satisfies all clauses. This Hamiltonian turns out to be a natural choice to solve $k$-SAT-type problems.

Our main result concerns the average number of unsatisfied constraints. Our adiabatic quantum machine can differentiate polynomially between $k$-SAT instances with no solutions and $k$-SAT instances having many solutions. We interpret the number of solutions in terms of the average number of unsatisfied constraints, which is a number between 0 and $m$, where $m$ is the number of clauses or interaction terms in the $k$-local Hamiltonian. If the average number is low enough (below 1/2), the time complexity will be polynomial in $n$. We show that such a condition on the average of unsatisfied constraints has some computational benefits.

Classically, one can differentiate between sets of constraints with no solutions and sets of constraints with many solutions using a random process, picking any assignment, and checking each of the constraints. This is also well known as an RP algorithm. The complexity class RP is the set of all decision problems $L$ for which there exists a probabilistic polynomial time Turing machine $A$ satisfying

$$x \in L \Rightarrow \Pr[A(x) = 1] > \frac{1}{2},$$
$$x \notin L \Rightarrow \Pr[A(x) = 1] = 0.$$ 

For example, $L$ could represent the property of ‘being prime’, and such a probabilistic algorithm will give the answer NO (0) if $x$ is not prime, and will give the answer YES (1) with high probability if $x$ is indeed prime. It could give the answer NO even if $x$ is prime, although this error will happen with low probability [8].

Our adiabatic quantum algorithm solves the problem of differentiating, up to any predetermined accuracy, in polynomial time and without the need of a random generator. Moreover, in case it is known in advance that there are many solutions, this algorithm can produce such a solution (up to any predetermined accuracy) in polynomial time. This is the adiabatic parallel of some well-known classical deterministic algorithms for the solution of formulae that have many satisfying assignments (see, for example, [9]).
In principle, one could simulate such classical algorithms, as in [9], by using quantum gates [7] and then construct an adiabatic quantum algorithm, as in [6]. The resulting effective Hamiltonian will realize the total history state of the original circuit as its ground state and will be hard to construct in terms of \(k\)-local operators. Instead, we use simple Hamiltonian operators, as in [1], which are quite natural for \(k\)-SAT problems. These algorithms could, in principle, be physically implemented in the near future.

From the point of view of computational complexity theory, this quantum algorithm is equivalent to a classical RP algorithm, both of which are polynomial. However, there are several motivations for presenting this quantum algorithm. Firstly, as mentioned above, this is an attempt to solve this problem directly by an adiabatic quantum computer by using very natural and simple Hamiltonians. Secondly, classically, to get high accuracy from an RP algorithm, one needs to operate the classical algorithm over and over again. Our adiabatic machine requires only one measurement. This is the outcome of the analytic proof of the polynomial asymptotic behavior of the corresponding gap between eigenvalues. Thirdly, this adiabatic machine realizes an example where the use of the gap between manifolds of eigenvectors can benefit computational theory.

The total time evolution \(T\) of an adiabatic computer is typically proportional to the inverse of the square of the minimal gap between the 0-eigenvalue (ground state) and the next closest (excited state) eigenvalue, over the entire evolution time interval \([0, T]\). In the following, we extend this concept by analyzing the gap between the 0-eigenmanifold and the next excited eigenmanifold (see also [2], [10]–[12]). We need this extension because, in our \(k\)-SAT problems, the lowest energy states may be degenerate at the end of the evolution, i.e. there could be many solutions that satisfy all clauses. We divide the unit time interval \(s \in [0, 1]\) into two subintervals \((s = t/T\) with \(t\) being real time). In the first one, for small time \(s\), we use Rayleigh’s minimax principles [13] to establish bounds on the gap. This is one of the reasons for realizing an initial Hamiltonian that is a positive definite operator. For the second subinterval, i.e. for \(s\) near 1, we use the Geršgorin disc theorem [14]. We then show that a condition on the average number of solutions can be imposed to ensure that the two subintervals cover the whole unit interval.

It seems reasonable to think that there could be other computational benefits from considerations of the gap between eigenmanifolds (as opposed to the gap between single states) of the final Hamiltonian when this concept is applied to other adiabatic quantum algorithms.

2. The quantum adiabatic machine

2.1. The state space

Let \(\mathcal{H}\) be a two-dimensional vector space over the complex numbers \(\mathbb{C}\), and \(\mathcal{H}^n = \mathcal{H} \otimes \mathcal{H} \otimes \cdots \otimes \mathcal{H}\) the tensor-product space of \(\mathcal{H}\), whose dimension is \(2^n\). Define the standard basis in \(\mathcal{H}\)

\[
|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix},
\]

and the rotated one

\[
|+\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad |\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}.
\]

New Journal of Physics 12 (2010) 075028 (http://www.njp.org/)
If $\sigma_z$ is the Pauli (spin-1/2) matrix along the $z$-direction, the operator

$$\tilde{\sigma}_z = \frac{1}{2}(1 - \sigma_z) = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

acts as $\tilde{\sigma}_z |\alpha\rangle = |\alpha\rangle$ on the $\sigma_z$ eigenbasis ($\alpha = 0$, 1 or FALSE, TRUE). Similarly, if $\sigma_x$ is the Pauli spin matrix along the $x$-direction, the operator

$$\tilde{\sigma}_x = 21 + \sigma_x = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$

acts as $\tilde{\sigma}_x |\mp\rangle = |\mp\rangle$ and $\tilde{\sigma}_x |\pm\rangle = 3|\pm\rangle$. By extension, Pauli-like operators acting on $\mathcal{H}^n$ are defined by simple tensor products:

$$\tilde{\sigma}_{z,x} = 1 \otimes 1 \otimes \cdots \otimes \tilde{\sigma}_{z,x} \otimes \cdots \otimes 1.$$  

2.2. The problem Hamiltonian

In complexity theory the satisfiability problem is a decision problem whose instance is a Boolean expression written in terms of the Boolean primitive functions AND, OR and NOT acting on a set comprising a finite number of variables. The question is to tell whether there exists an assignment of TRUE or FALSE to the variables such that the entire Boolean expression is true, or to prove that there is no such assignment. For example, consider the following formula with three variables and five clauses:

$$(x \lor y \lor z)(x \lor y)(y \lor z)(z \lor x)(x \lor y \lor z).$$

In the above example, each clause contains at most three variables and we say that the formula belongs to 3-SAT. It is well known that 3-SAT is enough to describe the whole set of NP. It is also well known that 2-SAT is a problem of polynomial complexity [15] (although classically the 2-SAT problem is easy, we are not aware of any quantum adiabatic algorithm that uses 2-local Hamiltonians that rigorously proves that it is in the class P for arbitrary instances; see also [16]).

Consider now a set of $m$ clauses $\{C_j\}_{j=1,...,m}$ in $n$ Boolean variables $\{x_i\}_{i=1,...,n}$, where $m$ is a polynomial in $n$ and its ratio is defined as $\beta = m/n$.

For computational problems inside SAT, it is easy to construct a Hamiltonian. Given a $k$-SAT clause $C$ in $n$ Boolean variables, we can define an energy function $h_C$ on $\mathcal{H}^n$ such that it is 0 whenever the clause is satisfied and 1 otherwise. For example, for a 3-SAT clause

$$C = x_1 \lor x_2 \lor x_3;$$

then the energy function is

$$h_C = (1 - \tilde{\sigma}^1_z)(1 - \tilde{\sigma}^2_z)(1 - \tilde{\sigma}^3_z).$$

Clearly, $h_C$ is 1 only when $|0\rangle^{\otimes 3}$ (UNSATISFIED) and 0 (SATISFIED) otherwise. In general, the following mapping for the Boolean variable and its negation results:

$$x_i \rightarrow 1 - \tilde{\sigma}^I_x, \quad \bar{x}_i \rightarrow \tilde{\sigma}^I_x.$$ 

Next, for a family of clauses we take the sum of the cost energies (which represents the formula in Boolean logic):

$$H_P = \sum_{j=1}^m h_{C_j}, \quad \text{with trace } \text{Tr} H_P = \sum_{r=1}^{2^n} t_r = T,$$
thus defining the problem Hamiltonian. Then, $H_P$ has eigenvalue 0 whenever $|\alpha_1\rangle \otimes \cdots \otimes |\alpha_n\rangle$ satisfies all clauses. Otherwise, each diagonal element of $H_P$ (in the standard basis), $t_r$, counts the number of unsatisfied clauses $m_r$ whose fraction is $0 \leq \gamma_r = m_r/m \leq 1$ ($\gamma_r = 0$ meaning that all clauses are satisfied). In other words, $H_P$ is a penalty function for unsatisfied clauses. Therefore, for the $k$-SAT problem, the equivalently mapped physical system corresponds to a $k$-local, or $k$-body, problem. Note that $H_P$ is a positive semi-definite operator (all eigenvalues are strictly non-negative).

2.3. The initial Hamiltonian

Now, consider the simple operator

$$H_B = \sum_{i=1}^{n} \tilde{\sigma}_i^i$$  \hspace{1cm} (10)

as the initial ($s = 0$) Hamiltonian. Note that $\tilde{\sigma}_i$ is a positive definite matrix, i.e. all its eigenvalues are strictly positive; therefore $H_B$ is also positive definite.

Finally, the following time-dependent Hamiltonian,

$$H(s) = (1-s)H_B + sH_P,$$  \hspace{1cm} (11)

represents the quantum algorithm. Observe that $H(s)$ is positive definite for $0 \leq s < 1$. Moreover, $H_B$ has a non-degenerate ground state, while $H_P$ may have a massively degenerate ground state manifold (many configurations have the lowest eigenvalue), which makes the use of the adiabatic theorem of quantum mechanics a bit subtle, and the relevant question that emerges is: Which gap is the relevant one?

2.4. The quantum adiabatic algorithm

Our Hamiltonian $H(s)$ evolves for a period of time that can be calculated in advance and is large enough to make sure its 0-eigenvector (at $s = 1$) is very close to a solution to the family of SAT clauses if there is such a solution. Then we stop the evolution and measure its outcome. We now check whether the final vector satisfies all clauses. This also takes a very short time. If indeed this vector is a solution, we then know that the answer is YES; otherwise our vector will not satisfy at least one clause and we can say with almost certainty that the answer is NO. Indeed, we can get an unsatisfying vector while the true answer is YES, but this should have a negligible probability. Our algorithm can reduce this probability to 0 for all practical purposes.

The evolution time of an adiabatic quantum machine is bounded from below by a fraction that depends on the minimal gap between the 0-eigenvalue and the next excited eigenvalue, taken over the whole time interval. To make sure this gap is not artificially increased by blowing up the Hamiltonian itself, we also demand that the matrix coefficient of the derivative of the Hamiltonian is bounded. Define $\Delta(s) = \lambda_2(s) - \lambda_1(s)$, where $\lambda_1(s)$ and $\lambda_2(s)$ are the ground and first excited state eigenvalues, with corresponding $|n_1(s)\rangle$, $|n_2(s)\rangle$ eigenvectors, and $\Delta_{\text{min}} = \min_{0 \leq s \leq 1} \Delta(s)$. In general, then, the minimal time we need to evolve the machine satisfies [1]

$$T > \frac{\delta^2}{\Delta_{\text{min}}}$$  \hspace{1cm} where $\delta = \max_{0 \leq s \leq l} |\langle n_2(s) | \frac{dH}{ds} | n_1(s) \rangle|$.  \hspace{1cm} (12)
For the algorithm presented here, $\Delta_{\text{min}}$ is related to the gap between the 0-eigenmanifold and the first excited eigenmanifold. Moreover, to be completely rigorous we have to maximize $\delta$ over all $|n_2(s)\rangle$ (respectively $|n_1(s)\rangle$) eigenvectors of the 1 (respectively 0) eigenmanifold (see also [10]).

3. The complexity of the algorithm: the evolution time

One can, in principle, determine the eigenvalues $\lambda_1(s)\{H(s)\}$ ($\ell \leq 2^n$) of the snapshot Hamiltonian $H(s)$, where $s \in [0, 1]$. Generically, the complexity of the adiabatic algorithm is a function of the gap between eigenvalues of $H(s)$ [2]. In the following we want to establish bounds among some of those eigenvalues, in particular, a lower bound on the gap $\Delta(s) = \lambda_2(s) - \lambda_1(s)$, where $\lambda_1(s)$ and $\lambda_2(s)$ are the ground and first excited eigenvalues, respectively. At $s = 0$, the ground state $|\rangle_1 \otimes \cdots \otimes |\rangle_0 = |\rangle \otimes \cdots \otimes |\rangle$ is non-degenerate. For small values of $s$, near 0, one can guarantee a gap between the first (ground) and the second (first excited) eigenvalues of $H(s)$, because of our choice of initial Hamiltonian. At large enough values of $s$, all eigenvalues that converge to 0 (at $s = 1$) get disconnected from the rest. We compute the values of $s$ that satisfy either of the conditions. We then show that the class RP is exactly the one that makes the two sub-intervals cover the unit interval.

Let us start by presenting the well-known Rayleigh’s minimax principles.

Theorem 3.1. (Rayleigh [13]) Let $\tilde{H}$ be a positive definite symmetric matrix. Then, the quotient

$$R_1[|\psi\rangle] = \frac{\langle \psi | \tilde{H} | \psi \rangle}{\langle \psi | \psi \rangle}$$

is minimized by the eigenvector $|\phi_1\rangle$ with the smallest eigenvalue $\lambda_1$, and its minimal value is $R_1[|\phi_1\rangle] = \lambda_1$. In general,

$$\lambda_1 \leq R_1[|\psi\rangle],$$

thus defining the usual upper bound on the ground state eigenvalue. Less known is the following inequality:

$$\lambda_2 \geq \min_{|\phi\rangle: \langle \phi | \phi \rangle = 0} \frac{\langle \psi | \tilde{H} | \psi \rangle}{\langle \psi | \psi \rangle} = R_2[|\phi\rangle]$$

for all $|\phi\rangle$, which provides a way of estimating the second eigenvalue of $\tilde{H}(s)$ without knowing the first one, $\lambda_1$. The condition $\langle \psi | \phi \rangle = 0$ represents a constraint on an arbitrary vector $|\psi\rangle$.

Obviously, $\lambda_2 = \max_{|\phi\rangle} R_2[|\phi\rangle]$, and $\lambda_1 \leq R_2[|\phi\rangle]$, which implies $\lambda_1 \leq R_2[|\phi\rangle] \leq \lambda_2$.

Lemma. The following inequalities are consequences of Rayleigh’s principle:

(i) $\lambda_2\{H(s)\} \geq (1-s)\lambda_2\{H_B\}$,

(ii) $\lambda_1\{H(s)\} \leq (1-s)\lambda_1\{H_B\} + s M$,

where $M = T/2^n$ (and $T$ is the trace of $H_B$, see also formula (9) here).

Proof. (i) For the right choice of $|\phi_2\rangle$, we have

$$\min_{|\psi\rangle: \langle \psi | \phi_2 \rangle = 0} \frac{\langle \psi | H_B | \psi \rangle}{\langle \psi | \psi \rangle} = \lambda_2\{H_B\}, \quad \text{and} \quad \min_{|\psi\rangle: \langle \psi | \phi_2 \rangle = 0} \frac{\langle \psi | H_B | \psi \rangle}{\langle \psi | \psi \rangle} \geq 0,$$
since $H_P$ is a positive semidefinite operator. That proves relation (i). As for (ii), first observe that

$$\lambda_1\{H(s)\} = \min_{|\psi\rangle} \left( (1-s)\frac{\langle \psi | H_B | \psi \rangle}{\langle \psi | \psi \rangle} + s \frac{\langle \psi | H_P | \psi \rangle}{\langle \psi | \psi \rangle} \right).$$  

(17)

Being a minimum, $\lambda_1\{H(s)\}$ is smaller than the right-hand expression for any choice of $|\psi\rangle$. So, choose the ground state of $H_B$, $|\tilde{\phi}_1\rangle = |\rangle^\otimes n$, and

$$\frac{\langle \tilde{\phi}_1 | H_P | \tilde{\phi}_1 \rangle}{\langle \tilde{\phi}_1 | \tilde{\phi}_1 \rangle} = M,$$

(18)

which is a positive number. Then,

$$\lambda_1\{H(s)\} \leq (1-s)\lambda_1\{H_B\} + s M.$$  

(19)

This concludes the proof of the lemma.

Using (i) and (ii) in the lemma, we now compute the gap. To this end, we want to determine the conditions on $s$ under which the following inequality,

$$\Delta(s) \geq (1-s)(\lambda_2\{H_B\} - \lambda_1\{H_B\}) - s M,$$

(20)

is satisfied. We also know that $\lambda_2\{H_B\} - \lambda_1\{H_B\}$ is 2. Then, the inequality reads

$$\Delta(s) \geq 2 - s(M+2).$$

(21)

Now, for $\epsilon$ a positive infinitesimal and $s \leq (2-\epsilon)/(2+M)$, we have

$$\Delta(s) \geq 2 - s(M+2) \geq \epsilon.$$  

(22)

So far we have proved that for small $s$ the gap is secured. We want now to look at the upper end of the unit interval, i.e. close to $s = 1$, where we use the Geršgorin discs theorem.

**Theorem 3.2.** (Geršgorin [14]) Let $\tilde{H} = [h_{ij}]$ be an $n \times n$ matrix, and for $1 \leq i \leq n$ let

$$R'_i(\tilde{H}) = \sum_{j=1, j \neq i}^{n} |h_{ij}|$$

(23)

denote the $i$th deleted absolute row sum of $\tilde{H}$. Then all the eigenvalues of $\tilde{H}$ are located in the union of $n$ discs:

$$\bigcup_{i=1}^{n} \{z \in \mathbb{C} : |z - h_{ii}| \leq R'_i(\tilde{H})\}.$$  

(24)

Moreover, if a union of $k$ of these discs forms a connected region that is disjoint from all the remaining $n - k$ discs, then there are precisely $k$ eigenvalues of $\tilde{H}$ in this region.

Let us now apply the above theorem to $H(s)$. In each row of the spin matrix $(1-s)H_B$ there are exactly $n$ non-zero (non-diagonal) entries and all have the value $1-s$. Therefore, the radius of each disc is $n(1-s)$. The discs are centered 1$s$ units apart (which is the difference between non-equal elements of $sH_P$). To make sure the discs are $\epsilon$ apart, we demand that

$$n(1-s) < \frac{\epsilon}{2}.$$  

(25)
As $s$ goes to 1, the discs’ radii become smaller and the distances between adjacent discs become bigger. This means that, from $s > (2n + \epsilon)/(1 + 2n)$, all eigenvalues gather inside separated discs.

So far we have secured both ends of the unit interval. As we go from 0 to 1, as long as $s < (2 - \epsilon)/(2 + M)$, $\Delta$ is bigger than $\epsilon$, and if we move slowly enough we will stay on the 0-eigenvector. Now, from $(2 - \epsilon)/(2 + M)$ up to 1, $\Delta$ might get close to 0. This means that we might jump to the next excited state. For such $s$’s ($s > (2 - \epsilon)/(2 + M)$), if we can make the gap between the manifolds big enough, it will not be possible to jump out of the 0-eigenmanifold (still we might jump in between roots of the 0-eigenmanifold). So we have to guarantee that the gap between the manifolds is secured even before any two roots in the 0-eigenmanifold can get too close to each other.

In other words, we have to guarantee that

$$
\frac{2 - \epsilon}{2 + M} \geq \frac{2n + \epsilon}{1 + 2n}.
$$

It is easy to see that, for $\epsilon = \frac{1}{n^2}$, the above inequality holds if

$$
M \leq \frac{1}{2n}.
$$

(assume that $n$ is big enough$^4$). So, if the average $M$ is small enough, the whole unit interval is secured by a gap of $1/n^2$.

**Corollary 3.1.** Given a $k$-SAT instance satisfying the following condition—either there are no satisfying assignments, or the average number of unsatisfied clauses is smaller than $1/2n$—the above adiabatic algorithm could differentiate between the two options in polynomial time.

---

$^4$ Inequality (26) holds if $(2 - \epsilon)/(2 + 2n\epsilon) \geq M$ or $(2 - 3\epsilon - 2n\epsilon)/(2n + \epsilon) \geq M$. Now, if $\epsilon = 1/n^2$, then $(2 - 3\epsilon - 2n\epsilon)/(2n + \epsilon) \geq 1/2n$. Therefore, if $\epsilon = 1/n^2$ and $M < 1/2n$, inequality (26) holds.
We now interpret the condition on the average in terms of the number of roots. It turns out that we can easily separate between the case where there are no roots and the case where there are many roots. The question is: How many?

**Corollary 3.2.** Given a $k$-SAT instance satisfying the following condition—either there is no satisfying assignment or there are at least $2^n (1 - (1/2nm))$ satisfying assignments—the above adiabatic algorithm could differentiate between the two options in polynomial time.

**Proof.** Observe that, to satisfy $M \leq 1/2n$, we need about $((2nm - 1)/2nm) 2^n$ places where the diagonal matrix elements of $H_P$ are 0, i.e. solutions satisfying all clauses, since

$$M \leq \frac{2^{nm-1}}{2^{nm}} 2^n \cdot 0 + \frac{1}{2^{nm}} 2^n \cdot m = \frac{1}{2n}. \quad (28)$$

So, if a large exponential number $((2nm - 1)/2nm) 2^n$ of $H(s)$’s diagonal elements equal 0, we can use the above adiabatic quantum machine to find one solution in polynomial time. Otherwise, if all $H(s)$’s diagonal elements are different from 0, our machine, having evolved the same amount of (polynomial) time, will yield some non-significant vector. $\Box$

**Example.** Consider the following simple 5-SAT problem in six variables and two clauses:

$$C_1 = x_1 \lor x_2 \lor x_3 \lor \overline{x_4} \lor x_5,$$

$$C_2 = x_2 \lor \overline{x_4} \lor x_4 \lor x_5 \lor x_6.$$  

It is easy to see that there are exactly four configurations that do not satisfy $C_1$ or $C_2$ (but satisfy at least one of them). Therefore, $M = 4/64 = 1/16$, which is smaller than $1/2n = 1/12$.

**Corollary 3.3.** Given a $k$-SAT instance satisfying the following condition—either there is no satisfying assignment or the average number of unsatisfied clauses $M$ is smaller than $1/2$—the above adiabatic algorithm could differentiate between the two options in polynomial time.

**Proof.** Consider the Hamiltonian $H_P^{\otimes l}$. This Hermitian operator will increase the number of variables $n$ by the small factor $l$ and reduce $M$ (exponentially) to $M'$. Clearly we need to satisfy

$$M' < \frac{1}{2ln}.$$  

This will happen for $l$ just larger than $2 \log(n)$. So, for such small $l$, the Hamiltonian $H_P^{\otimes l}$ will meet the above requirements (of corollary 3.1). $\Box$

4. Discussion

We would like to start with a very important remark. Using the Geršgorin disc theorem means that, from a certain point onward, all eigenvalues gather around separated discs. It could happen that more than one eigenvalue is in the disc centered at 0. Suppose $k_0$ is the 0-degeneracy of the problem. Then, by Geršgorin’s theorem, there are $k_0$ eigenvalues in the disc around 0 for all such $s$’s (the discs are centered around elements of $H_P$). Some of them may even get together exponentially fast as $n \to \infty$. The radius of the disc is decreasing to 0, so if the quantum machine jumps into one of these eigenvalues, it continues its way to 0. The next disc contains all vectors with eigenvalue around 1, and this disc is separated from the 0 disc by a finite gap, and so on.
To sum up, in the lower end of the unit interval we can actually prove that the gap is polynomially small. This simply comes from the choice of our initial Hamiltonian. At the upper end of the interval we prove that the eigenvalues are clustered in non-overlapping discs. The relevant gap is now the gap between the discs. In this way, we showed that the relevant gap to consider is the gap between eigenmanifolds and not simply between the ground state and an excited state \([2, 11, 12]\). This is so, since these types of problems have large degeneracies in the ground state of \(H_p\), and the standard adiabatic theorem of quantum mechanics is not designed to handle these cases.

Cook [17] showed the \(k\)-SAT with \(k \geq 3\) to be NP-complete. However, NP-completeness is a worst-case notion and it is interesting to analyze situations where the satisfiability problem becomes easily solvable. In that regard, we would like to mention the important fact that our work also proves analytically that there is a subclass of \(k\)-SAT \((k \geq 2)\) that can be solved with polynomial complexity with an adiabatic quantum machine. The subclass is defined by the average fraction of unsatisfied clauses \(\bar{\gamma} = \frac{M}{m}\) or the average number \(M\) of unsatisfied clauses. The number \(M\) is somehow related to \(\beta = m/n\). Reducing the number of clauses \(m\) reduces the constraints and increases the number of satisfying assignments. Also, if \(m\) is reduced, then for each non-satisfying assignment, the number of non-satisfied clauses is also reduced (the penalty number for such an assignment). Increasing the number of 0s and decreasing the penalty factors causes \(M\) to decrease. From a certain point onwards, the average, \(M\), will be small enough so as to satisfy the above condition, equation (27). So, one should expect a phase transition at a certain value of \(\beta\), from a mostly satisfying to a mostly unsatisfied phase, indicating the hardness of the problem instances. In this regard, our analytic results are consistent with numerical simulations (see, for instance, \([18]\)).

References

[1] Farhi E, Goldstone J, Gutmann S and Sipser M 2002 arXiv:quant-ph/0001106
[2] Kato T 1950 J. Phys. Soc. Japan 5 435
[3] Ambainis A and Regev O 2004 arXiv:quant-ph/0411152
[4] Farhi E, Goldstone J and Gutmann S 2002 arXiv:quant-ph/0201031
[5] van Dam W, Mosca M and Vazirani U 2001 Proc. 42nd FOCS p 279 arXiv:quant-ph/0206003
[6] Aharonov D, Kempe J, Lloyd S, van Dam W, Landau Z and Regev O 2004 arXiv:quant-ph/0405098
[7] Nielsen M and Chuang I 2000 Quantum Computation and Quantum Information (Cambridge: Cambridge University Press)
[8] Rabin M 1980 J. Number Theor. 12 128
[9] Hirsch E A 1998 A fast deterministic algorithm for formulas that have many satisfying assignments Logic J. IGPL 6 59–71
[10] Teufel S 2003 Adiabatic Perturbation Theory in Quantum Dynamics (Lecture Notes in Mathematics 1821) (Berlin: Springer)
[11] Avron J E and Elgart A 1999 Commun. Math. Phys. 203 445
[12] Avron J E, Seiler R and Yaffe L G 1987 Commun. Math. Phys. 110 33
[13] Strang G 1988 Linear Algebra and its Applications (Belmont, CA: Brooks/Cole)
[14] Horn R and Johnson C 1985 Matrix Analysis (Cambridge: Cambridge University Press)
[15] Aspvall B, Plass M and Tarjan R 1979 Inf. Process. Lett. 8 121
[16] Kempe J 2003 arXiv:quant-ph/0303081
[17] Cook S A 1971 Proc. 3rd Annu. ACM Symp. on Theory of Computing p 151 (New York: ACM Press)
[18] Kirkpatrick S and Selman B 2001 Statistical physics and computational complexity More is Different: Fifty Years of Condensed Matter Physics ed N Phuan Ong and R N Bhatt (Princeton, NJ: Princeton University Press) pp 331–339