

BACKGROUND KNOWLEDGE

1 Quantum Computing

Analogous to the digital bits used for classical computation, a quantum computer requires quantum bits, more commonly known as qubits, as the fundamental storage unit. However, unlike the classical bits which can acquire the value of either a 0 or a 1, a qubit state can be a linear superposition of the classical bits. A general single qubit state is given as:

$$|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle \quad (1)$$

where a_0 and a_1 are the complex amplitudes such that $|a_1|^2 + |a_0|^2 = 1$. According to the principles of quantum mechanics, the qubit state collapses to the states $|0\rangle$ or $|1\rangle$, with probability $|a_0|^2$ or $|a_1|^2$ upon measurement.

A qubit state can also be represented in terms of spin of a particle, whose internal angular momentum can take two values - $\hbar/2$ corresponding to spin-up state ($|\uparrow\rangle$), or $-\hbar/2$ corresponding to spin-down state ($|\downarrow\rangle$). In this notation, it is conventional to associate state $|0\rangle$ with $|\downarrow\rangle$ and state $|1\rangle$ with $|\uparrow\rangle$. Since there are only two values that the internal angular momentum can acquire, the total angular momentum associated with the particle is $S=1/2$ [Kristel and Hans, and their references]. The three components of the Pauli matrices corresponding to the spin-1/2 operator \mathbf{S} spanned by these states are:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

For N such qubits, the state is a tensor product of the single qubit states. Since each single state is described using two complex amplitudes, a state with N qubits requires $2^N = L$ coefficients for its description, i.e.

$$\begin{aligned} |\psi\rangle &= (a_{01} |0\rangle_1 + a_{11} |1\rangle_1) \otimes (a_{02} |0\rangle_2 + a_{12} |1\rangle_2) \otimes \dots \otimes (a_{0N} |0\rangle_N + a_{1N} |1\rangle_N), \\ &= a_{01}a_{02}\dots a_{0N} |00\dots 0\rangle + a_{01}a_{02}\dots a_{1N} |00\dots 1\rangle + a_{11}a_{12}\dots a_{1N} |11\dots 1\rangle, \\ &= a_0 |00\dots 0\rangle + a_1 |00\dots 1\rangle + \dots + a_L |11\dots 1\rangle, \end{aligned} \quad (2)$$

where $a_0 = a_{01}a_{02}\dots a_{0N}$, $a_1 = a_{01}a_{02}\dots a_{1N}$ and similarly, $a_L = a_{11}a_{12}\dots a_{1N}$.

The basis in equation (2) is known as the computational basis for quantum computation, and is more conveniently notated as $|0\rangle = |00\dots 0\rangle$, $|1\rangle = |00\dots 1\rangle, \dots, |L\rangle = |11\dots 1\rangle$. In this representation, equation (2) thus becomes

$$|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle + \dots + a_L |L\rangle. \quad (3)$$

Therefore, the Hilbert space spanned by N qubits is a $L=2^N$ dimensional space.

Since Pauli matrices form a basis for a vector in 2×2 space, any operator acting on a qubit can be expressed as a linear combination of Pauli matrices [Quantum Walks and search algorithms, Renato Portugal]. The action of a Pauli matrix, σ^α , where $\alpha \in \{x, y, z\}$ on the i^{th} of the N qubits is represented as σ_i^α . Since the state in equation (2) is a product state, the Pauli operator σ_j^α can act just on the j^{th} qubit, while the other qubits are acted upon by identity matrices $\mathbb{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}_{i \neq j}$.

There are two kinds of operations that will be required in the course of this thesis to carry out all the simulations. These are single and double qubit operations.

For a system containing N qubits, a single qubit operator is a matrix of order $2^N \times 2^N$. The action of such an operator on an N -qubit state is given by the product of the $2^N \times 2^N$ matrix on 2^N vector, which results in another 2^N vector. However, the computational resources required to carry out this multiplication grow exponentially with the number of qubits, and the approach becomes unsuitable even for 20 qubits, where approximately 16 Tb of memory is required to store the matrix [Kristel Hans paper]. A way out for the problem could be to analytically let the operator σ_j^α act on the j^{th} bit, i.e. by changing the coefficients a_{0j} and a_{1j} in equation (2) while keeping all the other a_{0i} and a_{1i} , for $i \neq j$, intact. In (the?) above notation, if

$$|\psi\rangle = a_{01}a_{02}\dots a_{0N} |00\dots 0\rangle + a_{01}a_{02}\dots a_{1N} |00\dots 1\rangle + a_{11}a_{12}\dots a_{1N} |11\dots 1\rangle, \quad (4)$$

and $|\psi'\rangle = \sigma_j^\alpha |\psi\rangle$, then

$$|\psi'\rangle = (a_{01}a_{02}\dots a_{0N})' |00\dots 0\rangle + (a_{01}a_{02}\dots a_{1N})' |00\dots 1\rangle + (a_{11}a_{12}\dots a_{1N})' |11\dots 1\rangle, \quad (5)$$

For σ_j^z , the new coefficients are obtained by reversing the signs of all coefficients of $|\psi\rangle$ where the j^{th} bit is 1. This means,

$$(a_{*1}a_{*2}..a_{0j}..a_{0N})' = a_{*1}a_{*2}..a_{0j}..a_{0N}, \quad (6)$$

and,

$$(a_{*1}a_{*2}..a_{1j}..a_{0N})' = -a_{*1}a_{*2}..a_{1j}..a_{0N} \quad (7)$$

where * is used to indicate that the bits in the corresponding positions remain unchanged.

Not Including this and the second qubit operation part?

2 Optimization Problems

Mathematically, an optimization problem comprises of a cost function, $f_0(x)$ involving N variables $x_1, x_2, x_3, \dots, x_N$, such that $x = (x_1, \dots, x_N)$. The goal, then, is to find the solution x_0 that optimizes the cost function (generally, energy), subjected to a certain number of constraints $f_i(x_1, \dots, x_k)$, each imposed to a set of k variables (k -SAT problems)[Convex Optimization, Stephen Boyd (book)]. These multiple constraints, however, can cause frustration in the system, which can give rise to many local minima. This makes the determination of the optimal solution harder [Optimization using quantum mechanics: quantum annealing through adiabatic evolution: Santoro and Tosatti].

For finding the optimal solution, the whole spectrum of the cost function needs to be explored. This requires that the system should be able to escape from a local minimum, if it gets trapped in one, during this course. The method of simulated annealing was therefore devised [Kirkpatrick S, Gelatt C D Jr and Vecchi M P 1983 Science 220 671], where adding thermal fluctuations to the cost function, keeps the system from getting trapped in the local minima. If the barrier potential diverges or becomes very high, this approach can no longer be helpful. However, the quantum tunnelling effect can still allow for the search of global minima of a function of many Boolean variables by escaping from the poor local minima (**Check**). It was in this spirit that the technique of quantum annealing was originally devised by B.Apolloni, C.Carvalho and D.de Falco in 1988 [Quantum stochastic optimization, Apolloni, Carvalho, de Falco].

3 Quantum Annealing

Quantum annealing was conceived with the idea of encoding the solution of the combinatorial optimization problem in the ground state of a quantum Hamiltonian [Tameem Albash, Lidar: Adiabatic Quantum Computation]. The cost function can then be mapped on to the Ising model of spins, making use of the external magnetic field h_i^z and the spin couplings J_{ij}^z of the model. Thus the optimization problem can be expressed in the terms of the following Hamiltonian:

$$H_P = - \sum_{i=1}^N h_i^z \sigma_i^z - \sum_{\langle i,j \rangle} J_{ij}^z \sigma_i^z \sigma_j^z \quad (8)$$

where σ_i^z denotes the z component of Pauli-spin matrix acting on the i^{th} spin, and the set $\langle i, j \rangle$ represents the set of pairwise couplings.

Although the Hamiltonian in equation (8) is a 2-local Hamiltonian, the solution for problems with higher order interactions ($k \geq 3$) can still be encoded in the ground state of the Ising Hamiltonian by combining σ_k^z terms into products, such that at each point there are just two terms effectively, but at the expense of imposing more constraints to the cost function. This can lead to a large overhead in the number of variables [Perspectives of quantum annealing :Methods and implementations]. For the present work, we shall confine ourselves to 2-SAT problems.

The recipe for the annealing algorithm consists of starting with an initial Hamiltonian H_I , whose ground state can be easily determined and realised. Most commonly used, is the transverse field Hamiltonian:

$$H_I = - \sum_{i=1}^N h_i^x \sigma_i^x \quad (9)$$

The ground state for H_I is therefore the uniform superposition state:

$$|\psi\rangle = \frac{1}{(\sqrt{2})^N} (|0\rangle + |1\rangle)_1 * (|0\rangle + |1\rangle)_2 * \dots * (|0\rangle + |1\rangle)_N \quad (10)$$

The Hamiltonian is then slowly swept towards the problem Hamiltonian, with the means of an annealing parameter, say s , defined as $s = t/T$, where t is the instantaneous time, and T is the total annealing time. The instantaneous Hamiltonian, $H(t)$, corresponding to the most straightforward annealing scheme, is then given by:

$$H(t) = (1 - s(t))H_I + s(t)H_P \quad (11)$$

Therefore, the instantaneous Hamiltonian transitions from the initial Hamiltonian, $H(t = 0) = H_I$ to the problem Hamiltonian, $H(t = T) = H_P$.

According to the **quantum adiabatic theorem**, the instantaneous state of the system stays close to the ground state of Hamiltonian $H(t)$, if one starts with ground state of the initial Hamiltonian and if the driving from the initial Hamiltonian to the problem Hamiltonian is slow enough for the given minimum energy gap, Δ_{min} , between the ground state and the first excited state of the Hamiltonian $H(t)$ for $t \in [0, T]$. [ADIABATIC QUANTUM COMPUTATION, Enej Ilievsk]. Mathematically, adiabatic theorem of evolution holds when

$$T \gg \Delta_{min}^{-2} \quad (12)$$

where T is the total annealing time. [Perspectives of quantum annealing: Methods and implementations]. Thus, the problem of finding the optimal solution reduces to the problem of solving the time dependent Schrödinger equation for the resulting $H(t)$ (equation 11):

$$i \frac{\partial}{\partial t} |\psi\rangle = H(t) |\psi\rangle \quad (13)$$

However, the process of evolution, starting from the trivial ground state of the initial Hamiltonian and going to a non-trivial ground state of the problem Hamiltonian, is associated with a quantum phase transition. For such a transition, the minimum gap Δ_{min} can be assumed to follow

$$\Delta_{min} \propto e^{-cN} \quad (14)$$

for a positive constant c and N number of spins. [Perspectives of quantum annealing: Methods and implementations]. Substituting equation (14) in equation (12),

$$T \gg e^{2cN} \quad (15)$$

i.e., the total annealing time required for the evolution to be adiabatic grows exponentially with the number of spins. **Find a bridging sentence saying that this renders it impractical.**

However, altering the annealing scheme can help overcome this difficulty of an exponential increase in the computational resources with the number of variables [Perspectives of quantum annealing: Methods and implementations, Adiabatic quantum computation: Tameem Albash, Quantum Adiabatic Evolution Algorithms with Different Paths: Farhi, Different strategies for Optimization Using the Quantum Adiabatic Algorithm: Farhi, Nonstoquastic Hamiltonians and quantum annealing of an Ising spin glass]. In this work we employ a third Hamiltonian, the Trigger Hamiltonian - H_T in the time dependent Hamiltonian $H(t)$ (11)[Farhi-Goldstein, Farhi (Different), Review, Nonstoquastic]. The trigger Hamiltonian is constituted by the transverse spin couplings J_{ij}^x . Moreover, H_T should vanish at both the start and end of the annealing process, so that one can still start with the easily realizable ground state of the initial Hamiltonian, and the resulting state of the problem Hamiltonian remains unaffected. The instantaneous Hamiltonian thus takes the form:

$$H(t) = (1 - s)H_I + g * s(1 - s)H_T + sH_P, \quad (16)$$

where parameter g controls the strength of the added trigger.

Furthermore, for this thesis we deal with two types of trigger Hamiltonians - The ferromagnetic trigger (F) and the anti-ferromagnetic trigger (A), given as

$$H_T^F = - \sum_{\langle i, j \rangle} \sigma_i^x \sigma_j^x, \quad (17)$$

and

$$H_T^A = + \sum_{\langle i, j \rangle} \sigma_i^z \sigma_j^z, \quad (18)$$

with the same pairwise coupling set $\langle i, j \rangle$ as for the problem Hamiltonian. **What else about these triggers?** As will be seen in the following chapters, adding these triggers alter the energy spectrum considerably, which in

turn affects the overlap of the final state with the ground state in different ways.

There can be yet another approach for increasing the overlap of the final state obtained from the annealing process, with the ground state of the problem Hamiltonian. As seen in (15) for problems with small minimum gap Δ_{min} , the annealing time required for the evolution to be adiabatic, so that the state of the system always stays close to the instantaneous ground state, can be very large. In such cases, even what otherwise looks like a reasonable time, might actually be short, causing the system to transit to the first excited state at the minimum gap anti-crossing. Since most of the amplitude of the state then lies in the first excited state, the overlap with the ground state might decrease considerably.

However, if one chooses much smaller annealing times, the state can leak to higher excited states of the spectrum, even before the minimum gap anti-crossing. As a result, the final state can have more chances of having a larger overlap with the ground state of the instantaneous Hamiltonian, than in the case of it closely following the first excited state with vanishing overlap with the ground state, because of shift of some amplitude of the wave function back to the ground state at the minimum gap. This observation has been confirmed in the work by [Farhi et al: Different Perspectives], where reducing the total annealing time in problems with very small success probabilities, increases the probability of finding the final state close to the ground state of the problem Hamiltonian.

The following sections focus on two approaches that have been used in this work to solve the time-dependent Schrödinger equation. Suzuki-Trotter product formula has been adopted to track the evolution of the state, and to compute the overlap of the final state with the already determined ground state of the problem Hamiltonian. The full diagonalization method, on the other hand, has been used to calculate the errors involved in using the Suzuki-Trotter approximation, and to determine the energy spectra and minimum gaps for $H(t)$ for $t \in [0, T_A]$, where T_A is the total annealing time.

4 Exact Diagonalization

This method consists of determining the eigenvalues and eigenvectors of the Hamiltonian matrix, which is a $L \times L$ ($2^N \times 2^N$) matrix, at each time step.

For constructing the Hamiltonian matrix at time step $t \in [0, T_A]$, all the basis vectors in the computational basis are acted upon by the instantaneous Hamiltonian, given by equation (16). The action of the Hamiltonian on the i^{th} basis vector corresponds to the i^{th} column of the Hamiltonian matrix, e.g. for $|\psi\rangle = (1, 0, \dots, 0)^T$, $H|\psi\rangle$ gives the first column of the Hamiltonian matrix.

The resulting matrix is then diagonalized to obtain the eigenvalues Λ , and unitary matrix of the eigenvectors, V . Since $V^\dagger H V = \Lambda$, the unitary evolution operator $U(t) = e^{-itH} = V e^{-it\Lambda} V^\dagger$.

This approach, however, has some serious limitations. Firstly, the memory requirement to store the Hamiltonian matrix grows exponentially with the number of qubits. Additionally, the full diagonalization takes $\mathcal{O}(2^{3N})$ floating-point operations [Kristol-Hans paper]. Therefore, this method is rendered impractical for solving the time dependent Schrödinger equation for systems more than 20 qubits.

5 Suzuki-Trotter Product Formula

Since solving the time dependent Schrödinger equation requires to compute unitary matrix exponentials, Lie-Trotter-Suzuki product formula can be used to construct the following unitary approximations:

$$U(t) = e^{-itH} = e^{-it(H_1 + \dots + H_K)} = \lim_{m \rightarrow \infty} \left(\prod_{k=1}^K e^{-itH_k/m} \right)^m. \quad (19)$$

For sufficiently small time step t , the first order approximation for $U(t)$ in (19) is

$$\tilde{U}_1(t) = e^{-itH_1} \dots e^{-itH_K}, \quad (20)$$

which holds good for $t \|H\| \ll 1$.

For improved accuracy, a second order approximation is made to $U(t)$ in (19), using $\tilde{U}_1(t)$ (20):

$$\tilde{U}_2(t) = \tilde{U}_1^\dagger(-t/2) \tilde{U}_1(t/2) = e^{-itH_K/2} \dots e^{-itH_1/2} e^{-itH_1/2} \dots e^{-itH_K/2}. \quad (21)$$

If $\tilde{U}_1(t)$ is unitary, so is $\tilde{U}_2(t)$. For unitary $\tilde{U}_2(t)$, the measure of error, calculated using the absolute difference between $U(t)$ and $\tilde{U}_2(t)$ grows cubically in t [Kristel-Hans paper, 62 from there], i.e.

$$\left\| U(t) - \tilde{U}_2(t) \right\| \leq ct^3 \quad (22)$$

for a positive constant c . If the whole annealing process requires n such time steps, the involved error becomes

$$\left\| U(t) - \tilde{U}_2(t) \right\| \leq nct^3. \quad (23)$$

Since $nt = T_A$, (23) is equivalent to

$$\left\| U(t) - \tilde{U}_2(t) \right\| \leq cnT_A t^2 = c't^2 \quad (24)$$

Substituting (??), (17 or 18) and (??) in (16), we obtain:

$$H(t) = (1-s) \left(- \sum_{i=1}^N h_i^x \sigma_i^x \right) + g * s(1-s) \left(\mp \sum_{\langle i,j \rangle} \sigma_j^x \right) + s \left(- \sum_{i=1}^N h_i^z \sigma_i^z - \sum_{\langle i,j \rangle} J_{ij}^z \sigma_i^z \sigma_j^z \right). \quad (25)$$

The Hamiltonian is then decomposed as follows:

$$H = H_{single} + H_x + H_z, \quad (26)$$

where $H_{single} = -(1-s) \left(\sum_{i=1}^N h_i^x \sigma_i^x \right) - s \left(\sum_{i=1}^N h_i^z \sigma_i^z \right)$, $H_x = \mp(1-s) \left(\sum_{\langle i,j \rangle} J_{ij}^x \sigma_i^x \sigma_j^x \right)$, and $H_z = -s \left(\sum_{\langle i,j \rangle} J_{ij}^z \sigma_i^z \sigma_j^z \right)$.

In general,

$$e^{i\mathbf{v} \cdot \boldsymbol{\sigma}} = \cos(v) \mathbb{1} + i \frac{\sin(v)}{v} \mathbf{v} \cdot \boldsymbol{\sigma}, \quad (27)$$

where $\mathbb{1}$ represents the identity matrix, and $v = \sqrt{v_x^2 + v_y^2 + v_z^2}$. Then,

$$e^{-itH_{single}} = \prod_{i=1}^N e^{it[(1-s) \sum_i h_i^x \sigma_i^x + s \sum_i h_i^z \sigma_i^z]} = \prod_{i=1}^N \begin{pmatrix} \cos(th_i) + i \frac{sh_i^z}{h_i} \sin(th_i) & i \frac{(1-s)h_i^x}{h_i} \sin(th_i) \\ i \frac{(1-s)h_i^x}{h_i} \sin(th_i) & \cos(th_i) - i \frac{sh_i^z}{h_i} \sin(th_i) \end{pmatrix} \quad (28)$$

where $h_i = \sqrt{(1-s(t))^2 h_i^{x^2} + s(t)^2 h_i^{z^2}}$.

The computational basis states are the eigenstates of the Pauli-z operator, σ_i^z . Thus e^{-itH^z} is a diagonal matrix in the computational basis, and its action on the input state changes the phase of each of the basis vectors. As H^z is a sum of pair interactions, it is trivial to implement this operation as a sequence of multiplications by 4×4 diagonal matrices [Kristel-Hans paper].

The same approach can be adopted for implementing H_x operations as well, by using the rotation operators Y_j as follows. Writing $Y = \prod_{i=1}^N Y_i$, we obtain

$$e^{-itH^x} = \bar{Y} Y e^{-itH^z} \bar{Y} Y = \bar{Y} e^{it \sum_{\langle i,j \rangle} J_{ij}^x \sigma_i^x \sigma_j^x} Y. \quad (29)$$