eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

ZIENTZIA
ETA TEKNOLOGIA
FAKULTATEA
FACULTAD
DE CIENCIA
Y TECNOLOGÍA

Gradu Amaierako Lana / Trabajo Fin de Grado
Ingenieritza Elektronikoko Gradua / Grado en Ingeniería Electrónica

A Quantum Model for Trend and Political Forecasting

Egilea/Autor/a:
Rubén Ibarrondo López
Zuzendaria/Director/a:
Prof. Iñigo L. Egusquiza
Zuzendarikidea/Codirector/a:
Dr. Mikel Sanz

Leioa, 2020ko ekainaren 17a /Leioa, 17 de junio de 2017

Contents

Introduction and objectives

Quantum adiabatic computation is a computational paradigm which employs the quantum adiabatic theorem in order to find configurations minimising a cost function (usually referred as energy), which codifies the solution to a certain problem. In particular, it has been shown that it can be applied for solving satisfiability problems [1], which are present in fields such as electronic design automation or scheduling problems. A satisfiability problem has several criteria that a given configuration should satisfy to solve the problem, or at least, satisfy as many conditions as possible. When the size of the problem scales up, finding that configuration is a very difficult task. Although the intervention of quantum phenomena is expected to increase the efficiency in solving such problems, some limitations must be considered. Therefore, we should keep in mind the constraints imposed by the physical implementation of quantum adiabatic processors, in order to know the scope of applicability of such resources. In this work we will revisit how quantum adiabatic computation can be employed to find the minimal energy configuration efficiently and we will review the limitations that must be considered.

The mathematical description of physical systems has inspired in the past models to describe social dynamics [2, 3, 4, 5]. Some of the models have actually been contrasted with experimental data, often showing that their results successfully replicate real social patterns. In this work, we will review a model for forecasting elections based on a spin system [6]. We will see that the spin couplings and individual magnetic fields can be used to describe interactions between the ideology of individuals and external agents. The orientation of each spin reflects a possible choice on a referendum and each interaction adds an energy contribution to the total energy of the system. The energy of the system can be minimised for a given configuration, which reflects the expected outcome for that referendum. We will describe the model and the experiment that was carried out to check the validity of its predictions. We will see how we obtained the data to build the model from the interactions of the individuals in social networks. Additionally, we show that this model can be implemented in a quantum adiabatic computer, providing an efficient method for solving it.

This work pursues three main objectives. The first one is to understand and explain quantum adiabatic computation, which requires mastering fundamental concepts on quantum mechanics and learning the practical steps to solve a problem with this technique. The second one is to see a particular example of the physical implementation of quantum computation, more precisely, the implementation with superconducting circuits. The third objective is to describe the procedure followed to develop and experimentally contrast the model in Ref. [6]. This required simulating quantum adiabatic processes and also getting familiar with the data retrieval using *Twitter* API, which was used to obtain the parameters of the model.

This work is divided into three chapters. In the first chapter, I review the necessary background to understand quantum adiabatic computation. In the second one, I explain the most relevant concepts about the implementation of quantum adiabatic computation with superconducting circuits. In the third chapter, I explain the model for forecasting elections and the experimental realization followed to contrast its results.

Chapter 1

Adiabatic quantum computation

Adiabatic quantum computation can be applied for solving satisfiability problems [1]. A satisfiability problem has several criteria that one wants to fulfil. The solution is obtained when a configuration that satisfies all the criteria is found, or at least, a configuration that fulfils as many conditions as possible. When the size of the problem scales up, finding that configuration is a very complex task. Such complexity could easily overcome the current capacity of classical computers, but it would be tractable for quantum computers.

The aim of this chapter is not to provide a detailed description of quantum mechanics and information, but to understand the principles of quantum processing units based on quantum adiabatic computation. However, I will revisit some fundamental concepts that we will employ. I will also explain some examples in order to understand how a problem can be adapted to be solved with this approach. This concepts will be key in the final chapter to solve a particular case of a forecasting model.

In Sec. 1.1, I introduce some relevant concepts concerning quantum mechanics, in particular, the concept of Hamiltonian. In Sec. 1.2, I review the adiabatic theorem which describes the physical principle in which quantum adiabatic computation relies on, this is mostly based on Ref. [1]. Finally in Sec. 1.3, I explain how quantum adiabatic evolution can be employed to perform quantum adiabatic computation to find solutions for given problems.

1.1 Fundamentals of quantum mechanics

In classical physics a state is a possible configuration of a system, described by the magnitudes that can be measured in it, which we call observables. For example, the position and velocity of a particle or the orientation of a compass with respect to a magnetic field. In classical physics we assume that there is a set of fixed numbers that are associated with those magnitudes, although we also assume there is an uncertainty in our measurement. In a quantum state we may have a well defined state, with certain values for the observables, but we may also have the state in a superposition of many possibilities, such as having a particle in a superposition of different positions.

For example, take a closed circuit where current can flow in clockwise or anticlockwise direction. We will not consider variations in the magnitude of the current, only in its direction. In a classical system we would only admit a state where the current flows either in one direction *or* the other, but in a quantum state we can have a superposition of both. If we use $|\circlearrowright\rangle$ to represent the state with current flowing clockwise and $|\circlearrowleft\rangle$ to represent the anticlockwise flowing current, then the most general state can be described by

$$|state\rangle = \alpha\,|\circlearrowright\rangle + \beta\,|\circlearrowleft\rangle\,, \tag{1.1}$$

where the coefficient of each state is known as its amplitude and they are complex numbers which have a modulus and a phase. The coefficient of each state contains the probability to measure it. Precisely, the probability to measure the current flowing clockwise, $|\circlearrowright\rangle$, is $|\alpha|^2$ and the probability to measure it anticlockwise, $|\circlearrowleft\rangle$, is $|\beta|^2$. From this property it can be inferred that the coefficients are complex numbers whose squared modulus must add up to 1, that is, $|\alpha|^2 + |\beta|^2 = 1$.

As a general definition, the *state* of a quantum system is the mathematical object from which we can obtain the probabilities associated with observables. Properties of quantum states open a range of possibilities for new paradigms of computation. For instance, in quantum information we use the quantum bit or *qubit*. A qubit is a two-state quantum system, where instead of having a 1 or a 0, as in a classical bit, we may have a superposition of both. There are many physical systems that can be used to implement the qubit, we will review some of them in Chapter 2.

A typical case where we must consider a quantum state is the spin of an electron. This magnitude describes the intrinsic angular momentum of the electron, and it is also related to its intrinsic magnetic momentum. It is as if the electron were a point magnet and the spin described its orientation. Due to quantum uncertainties we cannot define all the components of the spin vector simultaneously, which is why we usually define the spin in terms of up and down in a given direction. The state of an electron's spin can be generally described by

$$|\psi\rangle = \alpha\,|\uparrow\rangle + \beta\,|\downarrow\rangle\,, \tag{1.2}$$

where $|\uparrow\rangle$ and $|\downarrow\rangle$ refer to the state with spin upwards and the state with spin downwards.

The state, as shown below, can be represented by a complex vector containing the coefficients for each possible state. This must be done according to a defined basis, in which every state is represented by a two dimensional vector,

$$|\psi\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}, \quad \text{with} \quad |\uparrow\rangle = \begin{pmatrix}1\\ 0\end{pmatrix} \text{ and } |\downarrow\rangle = \begin{pmatrix}0\\ 1\end{pmatrix}. \tag{1.3}$$

In order to describe the state we may want to measure it. The probability to measure the spin upwards or downwards can be computed from the coefficients of the spin state, as we saw for the current flow state in Eq. 1.1. From those probabilities, we could compute the expected value for the direction of the spin, with $+1$ upwards and -1 downwards it would be $1 \cdot |\alpha|^2 + (-1) \cdot |\beta|^2$. This can be straightforwardly computed because we have described our state in terms of the upwards and downwards states. But in order to obtain a general description observables are represented by matrices, which represent operations in the quantum states. For example, the z direction of the spin can be described by the z Pauli matrix

$$\sigma_z = \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}. \tag{1.4}$$

Representing observables as operators, the expected value for a magnitude can be computed as

$$\langle\psi|\sigma_z|\psi\rangle = \begin{pmatrix}\alpha^* & \beta^*\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\begin{pmatrix}\alpha\\ \beta\end{pmatrix} = |\alpha|^2 - |\beta|^2, \tag{1.5}$$

where a^* refers to the complex conjugate of a. In particular the states that satisfy the eigenvector equation for an observable, $M|\psi\rangle = \lambda|\psi\rangle$, are called eigenstates of the operator and the expected value is the eigenvalue λ, which is obtained with certainty. For example, $|\uparrow\rangle$ and $|\downarrow\rangle$ represent the eigenvectors for σ_z with eigenvalues $+1$ and -1, respectively.

There are also defined operators for observables related with measuring the spin pointing in the x and y directions, which are σ_x and σ_y respectively

$$\sigma_x = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}, \text{ and } \sigma_y = \begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}. \tag{1.6}$$

And their expected values can be computed in a similar manner,

$$\langle\psi|\sigma_x|\psi\rangle = \begin{pmatrix}\alpha^* & \beta^*\end{pmatrix}\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \mathrm{Re}\{\alpha^*\beta\}, \tag{1.7}$$

$$\langle\psi|\sigma_y|\psi\rangle = \begin{pmatrix}\alpha^* & \beta^*\end{pmatrix}\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}\begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \mathrm{Im}\{\alpha^*\beta\}. \tag{1.8}$$

We can also obtain their eigenvectors and eigenvalues,

$$\text{for } \sigma_x : \quad \frac{|\uparrow\rangle + |\downarrow\rangle}{\sqrt{2}} \text{ with } +1, \text{ and } \frac{|\uparrow\rangle - |\downarrow\rangle}{\sqrt{2}} \text{ with } -1; \tag{1.9}$$

$$\text{for } \sigma_y : \quad \frac{|\uparrow\rangle + i|\downarrow\rangle}{\sqrt{2}} \text{ with } +1, \text{ and } \frac{|\uparrow\rangle - i|\downarrow\rangle}{\sqrt{2}} \text{ with } -1. \tag{1.10}$$

1.1.1 The Hamiltonian

The energy operator is called Hamiltonian, H. For instance, the Hamiltonian of an electron's spin in a magnetic field oriented in the z direction, $\mathbf{B} = B_0\mathbf{e_z}$, can be represented by

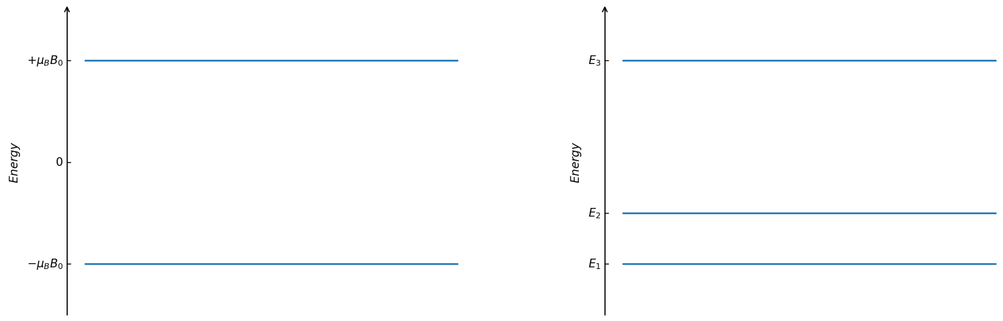
$$H = \mu_B B_0 \sigma_z = \mu_B B_0 \begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}, \tag{1.11}$$

with μ_B the Bohr magneton, which is a physical constant that represents the coupling of the electrons spin with the magnetic field. The value for the Bohr magneton is $\mu_B = e\hbar/2m_e$, where e is the electron charge, $\hbar$ the reduced Planck constant and m_e the electron mass. We notice that the eigenstates of this Hamiltonian are the states $|\uparrow\rangle$ and $|\downarrow\rangle$, with eigenvalues $\mu_B B_0$ and $-\mu_B B_0$. Those eigenvalues represent the energy of each state. Instead, if the magnetic field is oriented in the x direction then the Hamiltonian is $H = \mu_B B_0 \sigma_x$, with σ_x defined in Eq. 1.6. Although the eigenvalues are the same, now they correspond to the eigenstates of σ_x, shown in Eq. 1.9.

There are other physical systems where we could have more energy levels. For example, a 3 energy level Hamiltonian could be represented with a 3 dimensional matrix such us

$$H = \begin{pmatrix}E_1 & 0 & 0\\ 0 & E_2 & 0\\ 0 & 0 & E_3\end{pmatrix}. \tag{1.12}$$

The notation related to this should be settled. The eigenvalues of the Hamiltonian form the *spectrum of the Hamiltonian*, it is sometimes represented with a level graph as shown in Fig. 1.1 for two example Hamiltonians. The state with the lowest energy is called the *ground state*. The next energy level in the spectrum is known as the *first excited energy level*, and the following one as the second excited energy level, and so forth. When there is more than one state corresponding to an energy level, we say that the energy level is *degenerate*. If the energy levels are non-degenerate, then we can easily assign the index of the energy level to the corresponding state. In this case the, nth state refers to the state in the nth energy level.



(a) (b)

Figure 1.1: Energy levels for (a) Hamiltonian in Eq. 1.11 and (b) Hamiltonian in Eq. 1.12 for some energy values.

1.2 The adiabatic theorem

The meaning of the word adiabatic is discipline dependent. In the thermodynamic sense, a process is called adiabatic if during the process there is no heat transfer between the system and the environment. For example, take a gas in a cylinder with a moving wall controlled by a piston. If the piston is pushed strongly, so that the gas is rapidly compressed, it is assumed that the walls of the cylinder would keep heat from being transferred, because the process happens in a very sort time.

However, in quantum theory the term adiabatic process usually refers to a process that was performed very slowly. We will see a quantum example below, but a classical analogy can be described also considering a gas in a cylinder. Instead of pressing the piston strongly, we will apply a slowly increasing force. For example, placing small weights on the piston. As each weight applies a very small change, the force of the accumulated masses will instantaneously be compensated by the increase of the inner pressure of the gas. Thus, during an adiabatic process the system transits from the initial state to the final one through equilibrium states. In the quantum sense this would be called adiabatic even if the walls would not adiabatically isolate the gas. An example from daily life can be any of us carrying a dish filled with soup, walking very slowly while praying for the soup to remain in the equilibrium state and not spilling it.

From now on, we will work with the second definition for adiabatic process, so whenever the term adiabatic process is used it must be understood in the quantum sense, that is, a process where the physical system varies slowly.

The adiabatic theorem states that if a quantum system is initially in the nth eigenstate of an initial Hamiltonian, H_i, and this Hamiltonian is adiabatically transformed into a final Hamiltonian, H_f, then the state will evolve adapting to the new Hamiltonian's corresponding nth eigenstate. This is ensured as far as there are no energy levels crossing with the nth energy level. In particular, if the state is initially in the ground state, it will be in the ground state of the final Hamiltonian.

Let me now review the adiabatic theorem with a notation that can be applied for a general system, for a precise example look at Section 1.2.1. Along an adiabatic process the Hamiltonian will depend on the time t along the process, that is, $H = H(t)$. We will give the name T to the time required by the system to go from the initial Hamiltonian to the final one, i.e. the duration of the adiabatic process. Of course, the time evolving Hamiltonian must obey $H(0) = H_i$ and $H(T) = H_f$.

The evolution can be parametrized by $s = t/T$, with $0 \leq s \leq 1$. Assuming the following equation for $H(s)$ is satisfied for any value of the evolution parameter

$$H(s) \left| n, \psi(s) \right\rangle = E_n(s) \left| n, \psi(s) \right\rangle , \tag{1.13}$$

where $E_n(s)$ represents the energy of the nth eigenstate and $\left| n, \psi(s) \right\rangle$ is the corresponding eigenstate, both depending on the evolution parameter s. We will focus on the evolution of the ground state. In order to apply the adiabatic theorem, it is required the absence of level crossing, i.e. $E_0(s) < E_n(s)$ for all $n \neq 0$ and for $0 \leq s \leq 1$.

Now we can define a reference to say that the system is *slowly* changing. In fact, the length of the gap between the ground state and the first excited state imposes a limit in the time that an adiabatic process can take. The longer the process take, the slower the evolution of the system is done. The time required is constrained to,

$$T \gg \frac{\xi \hbar}{g_{\min}^2}, \tag{1.14}$$

where

$$\xi = \max_{0 \leq s \leq 1} \left| \left\langle n = 1; s \left| \frac{\mathrm{d}H}{\mathrm{d}s} \right| n = 0; s \right\rangle \right| \quad \text{and} \quad g_{\min} = \min_{0 \leq s \leq 1} (E_1(s) - E_0(s)). \tag{1.15}$$

The parameter $g_{\min}$ refers to the minimum energy gap over ground state, during the evolution. In particular, if there exists a value of s for which $E_1(s) - E_0(s) \to 0$, then $g_{\min} \to 0$ and $T \to \infty$ which shows that in the limit where the gap is too small the adiabatic process would require an infinite time.

The parameter ξ is the maximum value matrix element of the operator $\frac{\mathrm{d}H}{\mathrm{d}s}$ between the first excited state ($n = 1$) and the ground state ($n = 0$). We will not go into a precise description, but usually this parameter is in the order of a typical eigenvalue of H, and its effect in the constraint imposed to the parameter T can be neglected compared to the effect of the energy gap.

For a finite time T, the adiabatic transition from the initial state to the final one may not be perfect. The error of the adiabatic process, ε, can be defined as the component of the final state that is orthogonal to the final ground state. In many cases, the error bound can be approximated by [7]

$$\varepsilon \lesssim \frac{\hbar}{T} \max_s \frac{\left\| \frac{\mathrm{d}}{\mathrm{d}s} H(s) \right\|}{\left| E_1(s) - E_0(s) \right|^2}, \tag{1.16}$$

The tolerance in the error for the adiabatic process imposes a lower limit for the evolution time. Nevertheless, the evolution time cannot be increased indefinitely, the limit is usually imposed by the coherence time of the physical system. After the coherence, time the quantum state collapses to a classical thermal of state, loosing its quantum properties. This feature is explained in Chapter 2.

1.2.1 Example of the adiabatic theorem

As an example of an adiabatic process, let us take the case of an electron's spin initially in a magnetic field in z direction in its ground state, downwards. We will progressively rotate the direction of the magnetic field until it is in the x direction. The adiabatic theorem states that if the transition $B_z\mathbf{e_z} \to B_x\mathbf{e_x}$ is performed smoothly, the electron's spin will remain in the ground state, aligning its direction with the opposite direction of the magnetic field.

The Hamiltonian for the process can be described by

$$H(s) = -B_z\mu_B(1-s)\sigma_z - B_x\mu_B s\sigma_x = -\mu_B \begin{pmatrix} B_z(1-s) & B_x s \\ B_x s & -B_z(1-s) \end{pmatrix}, \tag{1.17}$$

the Hamiltonians for the magnetic fields in direction z and x were described in Sec. 1.1.1. It can be computed that the eigenvalues for $H(s)$ are

$$E_0(s) = -\mu_B\sqrt{B_z^2(1-s)^2 + B_x^2 s^2} \tag{1.18}$$

$$E_1(s) = +\mu_B\sqrt{B_z^2(1-s)^2 + B_x^2 s^2}. \tag{1.19}$$

In Fig. 1.2 the evolution for the both energies is shown for the case $B_z = B_x = B_0$ and the initial and final states are noted. These states are the eigenvectors of σ_z initially and the eigenvectors of σ_x finally. According to the adiabatic theorem, if the system is initially in the ground state $|\uparrow\rangle$ then it will finally be in the new ground state $(|\uparrow\rangle + |\downarrow\rangle)/\sqrt{2}$.

In this case the upper bound for the error can be calculated from Eq. 1.16. For the term $\left\| \frac{\mathrm{d}}{\mathrm{d}s} H(s) \right\|$ we have

$$\frac{\mathrm{d}}{\mathrm{d}s} H(s) = -\mu_B B_0 \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix} \Rightarrow \left\| \frac{\mathrm{d}}{\mathrm{d}s} H(s) \right\| = \sqrt{2}\mu_B B_0, \tag{1.20}$$

which does not depend on s. And the term of the energy gap is

$$\left| E_1(s) - E_0(s) \right|^2 = 4\mu_B B_0^2(1 + 2s^2 - 2s), \tag{1.21}$$

which is minimised for $s = 1/2$ with the value $\min_s \left| E_1(s) - E_0(s) \right|^2 = 2\mu_B^2 B_0^2$. Then, the expected error with an adiabatic time T is

$$\varepsilon \lesssim \frac{\hbar}{\sqrt{2}\mu_B B_0} \frac{1}{T}. \tag{1.22}$$

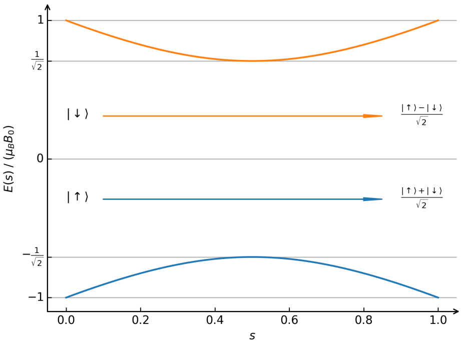


Figure 1.2: Evolution of the two energy levels of Hamiltonian in Eq. 1.17, with $B_z = B_x = B_0$.

1.3 Computation based on adiabatic evolution

Many problems can be represented by a cost function which we want to minimise under certain constraints. For example, if we were trying to build a low noise amplifier, we would be looking for the best combination of electronic devices minimising the noise, ensuring at least a minimum amplification power. The idea of quantum adiabatic computing is to map our cost function to the Hamiltonian of a quantum system, and use the adiabatic theorem to find the minimum of that function.

The system is initialized in a Hamiltonian whose ground state can be straightforwardly obtained, named H_0. Starting from that ground state the system is slowly driven to the Hamiltonian representing the cost function of the problem, H_P. Finally, according to the adiabatic theorem, the system will be in the ground state of H_P, which encodes the desired configuration. The Hamiltonian of the evolution can be written as

$$H(t) = \left(1 - \frac{t}{T}\right) H_0 + \frac{t}{T} H_P. \tag{1.23}$$

We will illustrate how to obtain H_P for a simple problem, afterwards we will describe a more general procedure to build a problem Hamiltonian.

Two-puppet dilemma. Imagine that we have three children (1, 2 and 3) and two kind of puppets, the Teddy Bear and the Rainbow Unicorn. The problem stems from trying to fulfil all of their whims: 1 and 3 both would like to have the same kind of puppet, 2 and 3 would also like the same, but 1 and 2 are not getting on well lately so they definitely do not want the same puppet. You prefer to buy the Teddy Bear, as the Unicorn is quite expensive. Which is the choice minimising the nonconformity? We will see how we can build a H_P representing the nonconformity as a cost function.

The nonconformity can be represented by the energy of a system, where each children would be a quantum spin. The two states of the spins tell the puppet each one would obtain, the state $|\uparrow\rangle$ represents the Teddy Bear and $|\downarrow\rangle$ the Rainbow Unicorn. We will build the Hamiltonian defining separate terms which consider each condition as a contribution to the overall energy.

The preference to buy the cheapest puppy can be represented by a magnetic field applying a *bias* in the three spins, which leads to the Hamiltonian

$$H_{\text{bias}} = -B_0\sigma_z^{(1)} - B_0\sigma_z^{(2)} - B_0\sigma_z^{(3)}, \tag{1.24}$$

where $\sigma_z^{(i)}$ represents the z-direction operator applied in the ith spin and $B_0 > 0$. The ground energy of this term is obtained for the case wherein the three children get the Teddy Bear, which is described by three spins up $|\uparrow\uparrow\uparrow\rangle$ with energy $-3B_0$.

Now we introduce the fact that 3 wants the same puppet as their siblings introducing a *coupling* between their spins represented by

$$H_{\text{coupling\_1}} = -J_{13}\sigma_z^{(1)}\sigma_z^{(3)} - J_{23}\sigma_z^{(2)}\sigma_z^{(3)}, \tag{1.25}$$

with $J_{ij} > 0$, which is minimized if 1 and 3 are aligned and 2 and 3 too. Both $|\uparrow\uparrow\uparrow\rangle$ and $|\downarrow\downarrow\downarrow\rangle$ have the minimum energy $-(J_{13} + J_{23})$ for this term.

Similarly, we can represent the condition of 1 and 2 not admitting to have the same puppet with a coupling that promotes counter-alignment

$$H_{\text{coupling\_2}} = J_{12}\sigma_z^{(1)}\sigma_z^{(2)}, \tag{1.26}$$

which is minimised for any state where they are not aligned, no matter the direction of 2, which are $|\uparrow\downarrow\uparrow\rangle$, $|\uparrow\downarrow\downarrow\rangle$, $|\downarrow\uparrow\uparrow\rangle$ and $|\downarrow\downarrow\uparrow\rangle$, with energy $-J_{12}$.

If we combine the three terms, we get the Hamiltonian for the problem

$$H_P = H_{\text{bias}} + H_{\text{coupling\_1}} + H_{\text{coupling\_2}}. \tag{1.27}$$

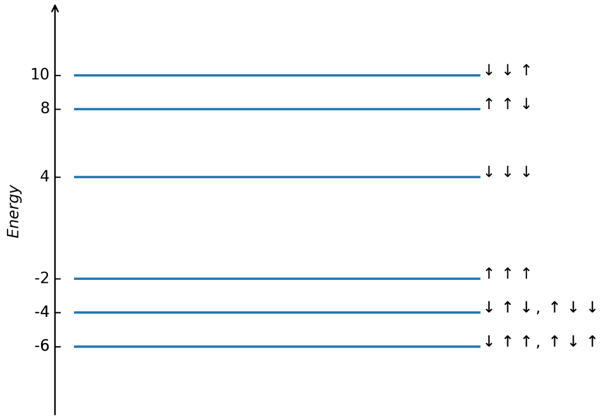


Figure 1.3: Energy levels for the two puppet dilemma, the states corresponding to each level are noted in their right.

We will set values for the constants in order to obtain a particular solution. We set the overall bias to $B_0 = 1$ and the aligning coupling to $J_{13} = J_{23} = 2$. As we want to really avoid the case where 1 and 2 obtain the same puppet we will set the second coupling to $J_{12} = 5$, so that it is higher than the other two. With this values, the spectrum of H_P is represented in Fig. 1.3. We can see that there are two states that correspond to the lowest energy levels, with energy -6: The $|\downarrow\uparrow\uparrow\rangle$ and $|\uparrow\downarrow\uparrow\rangle$, which means that the choices that lower the cost function are to buy either the Teddy Bear to 1 and 2 or to 2 and 3, with the remaining child getting the Rainbow Unicorn.

From the two-puppet example we can deduce that a problem Hamiltonian can be built employing many terms to represent each condition that we want to fulfil. In particular, we have seen that a preference can be represented by a magnetic bias, B_i, as we did to introduce our preference to buy the cheapest puppet in Eq. 1.24. Similarly, using coupling terms, J_{ij}, such as in Eq. 1.25 and Eq. 1.26, we can represent the tendency towards alignment or counter-alignment. Additionally, we have described a numerical example where we have employed the amplitude of the biases and couplings to assign different weights to each condition, such as choosing the counter-alignment term bigger than the others.

The Hamiltonian obtained in the two-puppet dilemma (Eq. 1.27) is a particular case of a quantum Ising Hamiltonian. The parameters B_i are known as individual biases and J_{ij} are the coupling constants. For a system with n spins we have

$$H(B_i, J_{ij}) = \sum_i B_i \sigma_z^{(i)} + \sum_{i<j} J_{ij} \sigma_z^{(i)} \sigma_z^{(j)}, \tag{1.28}$$

where the second sum is performed for every i and every j greater than i. This structure for a Hamiltonian can be useful to implement the H_P of many different problems. We can also obtain the classical Ising Hamiltonian employing the variables $s_i = \pm 1$ for each classical spin

$$\mathcal{H}(B_i, J_{ij}) = \sum_i B_i s_i + \sum_{i<j} J_{ij} s_i s_j. \tag{1.29}$$

This allows us to straightforwardly show why adiabatic computing is particularly promising for *satisfiability problems.* This problems arise when we must find a configuration that satisfies many Boolean conditions. That is, we have a system of n bits and we are given a set of Boolean clauses for them. In general, it is easy to find a configuration satisfying one of these clauses, but finding one which satisfies all of them is challenging. Additionally some of the conditions are incompatible, thus finding a perfect solution is impossible and the configuration satisfying as many clauses as possible is sought. In these cases, a weight can be assigned to each constraint, which allows us to prioritise some constraints over others.

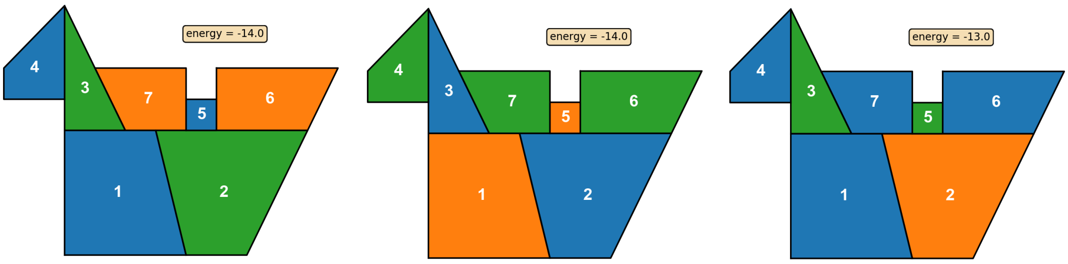
We can define the clauses so that we can add their result and construct an overall energy (cost) function for the system, $\mathcal{H}$, as we did for the Ising Hamiltonian. The value of each bit can be related to a classical spin variable as $q_i = (s_i + 1)/2$ which is either 1 or 0. Let us consider a 2-satisfiability problem (2-SAT), where each Boolean clause can only involve at most 2 bits, the energy in that case is

$$\mathcal{H} = \sum_i \sum_{j \geq i} a_{ij} q_i q_j, \tag{1.30}$$

where the constants a_{ij} are the weight constants for each clause. If there is no constraint affecting the pair $q_i q_j$ then the weight can be set to $a_{ij} = 0$. If a constraint is to be applied on a single bit, then by setting $i = j$ we get $a_{ii} q_i q_i = a_{ii} q_i$, as q_i is a Boolean variable.

We will describe an example where the 2-SAT formulation is used to build a H_P. We will employ bit arrays to encode the possible configurations and describe how the values for each coupling constants can be decided.

Map colouring. Imagine that we want to colour regions in a map without using the same colour for adjacent regions employing a restricted amount of different colours. This simple formulation can derive into a difficult problem when the number of regions is big and there are scarce colours available. We will consider a small example with 7 regions and 3 different colours.



(a) (b) (c)

Figure 1.4: Output example of a map colouring problem. (a) and (b) show correct outputs with no colour match between adjacent regions, while (c) represents an unsuccessful output.

	1			2			3			4			5			6			7		
1	-2	3	3	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0
		-2	3	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
			-2	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
2				-2	3	3	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0
					-2	3	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0
						-2	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1
3							-2	3	3	1	0	0	0	0	0	0	0	0	1	0	0
								-2	3	0	1	0	0	0	0	0	0	0	0	1	0
									-2	0	0	1	0	0	0	0	0	0	0	0	1
4										-2	3	3	0	0	0	0	0	0	0	0	0
											-2	3	0	0	0	0	0	0	0	0	0
												-2	0	0	0	0	0	0	0	0	0
5													-2	3	3	1	0	0	1	0	0
														-2	3	0	1	0	0	1	0
															-2	0	0	1	0	0	1
6																-2	3	3	0	0	0
																	-2	3	0	0	0
																		-2	0	0	0
7																			-2	3	3
																				-2	3
																					-2

Figure 1.5: Couplings and biases for the map colouring problem. Each column/row delimited by lines represents a region and each contains three bits. The values in the diagonal are the individual biases and the off diagonal terms represent the couplings between different bits.

For this problem, we will use 21 bits, $q_i = 1$ or 0 with $1 \leq i \leq 21$. Each of the 7 regions will have three bits, each bit representing a possible colour it could have. Setting a bit to 1 means that the colour it represents is used in the region it is within. So if we get 100 in a given region it would represent the colour blue, 010 orange and 001 green.

To define the problem Hamiltonian, we will use the Hamiltonian defined in Eq. 1.30, so we have to define the values for the coupling constants a_{ij}. In order to promote colouring, $q_i = 1$ states, the individual bias of each bit will be set to $a_{ii} = -2\ \forall i$.

Nevertheless, it is important that within each region only one bit is set to 1, otherwise the colour encoding would be invalid. Thus, we set the coupling for the bit pairs (i, j) within the same region to $a_{ij} = 3$. Once the fundamental constraints are stated, we introduce our desired result imposing a coupling $a_{ij} = 1$ between the same color in adjacent cells. An example for region distribution is shown in Fig. 1.4, and the couplings definition for that case are shown in Fig. 1.5.

Figure 1.4 also represents three typical outcomes of a simulated adiabatic evolution process. Most of the outputs are valid solutions to the problem, which have energy -14. Whereas some solutions with some equal adjacent colours were also obtained, with energy -13, due to the small energy difference between both cases. However, no solutions with invalid colours were obtained owing to the energy penalty imposed to those cases.

In contrast to the two-puppet dilemma, where there are only 8 different configurations, for the map colouring problem there are 2^{21} possible configurations ($3^7 = 2187$ if we only consider the ones with a valid colour encoding). This shows how rapidly the complexity of a 2-SAT problem can grow and the possibilities emerging from the usage of adiabatic quantum computation. In fact, for this example we had to set only 55 non-zero coupling constants, which is an straightforward task compared to computing all the possible configurations. Adiabatic quantum computation enables an effective way to obtain a configuration minimising the cost function employing a H_P defined with the coupling constants, without having to compute the energy of every configuration.

Chapter 2

Implementation in superconducting platforms

A bit is the fundamental unit of classical information, which can take the values 1 or 0. It can be electronically implemented easily with transistors and two voltage levels. Its quantum counterpart, the qubit, must also be physically implemented if we desire to employ its promising properties. In this chapter, we will review how qubits can be designed in superconducting platforms and how to implement a quantum adiabatic process in this technology.

Firstly we will introduce some concepts about the Hamiltonian description of circuits, which are necessary for the quantization of circuits involving superconductors. Then, we will explain Josephson junctions, which are non-linear circuit elements used to build the superconducting qubit. Making use of these elements, we will construct a flux qubit. There are other qubit types, but we will focus on the flux qubit since this is the kind of qubit employed by D-Wave [8], which is a quantum computing company that performs quantum adiabatic computations on demand. We will briefly review some technical details of their computers [9], including the connectivity of the qubits, the topology of the architecture employed and an overview of the quantum processing units operation cycle.

2.1 Hamiltonian description of circuits

In order to describe the quantum behaviour of a superconducting circuit, let us introduce the fundamentals of the Hamiltonian description of circuits. In this Section, we will focus on linear elements and their quantum description. In the following Section, we will describe a non-linear element known as the Josephson junction. In any case, we will only focus on two-terminal elements.

An electronic circuit can be represented by a network of nodes connected by branches. For every time t, each node has an electrical potential which is given with respect to to a *ground node* with zero value. This allows us to define the voltage difference in a branch, $v_b(t)$, and the current through it, $i_b(t)$. The positive direction of these magnitudes are shown in Fig. 2.1. Two-terminal elements are circuit elements that connect only two nodes of the circuit, thus involving a single branch. Two-terminal linear elements can be theoretically described by the relation between $v_b(t)$ and $i_b(t)$. For the resistance in Fig. 2.1 we have the Ohm law $v_b(t) = R \cdot i_b(t)$, where the constant R is the resistance that depends on the particular device.

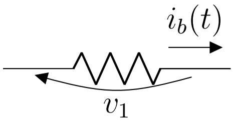


Figure 2.1: Example of a two-terminal element (a resistance) with the positive directions of the current through the branch $i_b(t)$ and the voltage difference $v_b(t)$.

Employing this variables it is possible to calculate the energy of the branch. Recalling that the power of a branch is obtained from $i_b(t)v_b(t)$, its energy

$$\varepsilon_b(t) = \int_{-\infty}^{t} i_b(t')v_b(t')\,\mathrm{d}t'\,. \tag{2.1}$$

Other basic circuit elements are the inductor and the capacitor, which respectively satisfy the equations

$$i_b(t) = \frac{1}{L}\int_{-\infty}^{t} v_b(t')\,\mathrm{d}t' \quad \text{and} \quad v_b(t) = \frac{1}{C}\int_{-\infty}^{t} i_b(t')\,\mathrm{d}t'\,, \tag{2.2}$$

where the constants L and C, called inductance and capacitance, depend on the particular device.

In superconducting circuits, it is a commonplace to describe the elements in terms of the branch flux, $\phi_b(t)$, and the branch charge, $Q_b(t)$, which are defined as

$$\phi_b(t) = \int_{-\infty}^{t} v_b(t')\,\mathrm{d}t'\,, \quad \text{and} \quad Q_b(t) = \int_{-\infty}^{t} i_b(t')\,\mathrm{d}t'\,, \tag{2.3}$$

where it is assumed that the circuit was initially (at $t \to -\infty$) at rest and there was no dissipation. This parameters allow us to describe the equations for the inductance and the capacitance in a simpler manner:

$$i_b(t) = \frac{\phi_b(t)}{L}, \quad \text{and} \quad v_b(t) = \frac{Q_b(t)}{C}. \tag{2.4}$$

The energy for these elements can be obtained from Eq 2.1. For the inductor

$$\varepsilon_L(t) = \int_{-\infty}^{t} \frac{1}{L}\phi_b(t')\frac{\mathrm{d}\phi_b(t')}{\mathrm{d}t'}\,\mathrm{d}t' = \frac{\phi_b^2(t)}{2L}, \tag{2.5}$$

and for the capacitor

$$\varepsilon_C(t) = \int_{-\infty}^{t} \frac{1}{C}Q_b(t')\frac{\mathrm{d}Q_b(t')}{\mathrm{d}t'}\,\mathrm{d}t' = \frac{Q_b^2(t)}{2C}. \tag{2.6}$$

Now, I will introduce the application of the Hamiltonian formulation to electrical circuits, which will lead to the quantization of the system [10]. Recalling that in the classical formulation the Lagrangian function is defined as de difference between the kinetic and the potential energies

$$\mathcal{L} = \mathcal{T} - \mathcal{V}, \tag{2.7}$$

both components are written in terms of the generalized coordinates, $q_\nu(t)$, where ν represents each degree of freedom, and their corresponding momenta, $p_\nu(t)$, defined as

$$p_\nu = \frac{\partial \mathcal{L}}{\partial \dot{q}_\nu}, \tag{2.8}$$

where $\dot{q}_\nu = \frac{\mathrm{d}q_v}{\mathrm{d}t}$. Then, the Hamiltonian of the system is obtained by

$$\mathcal{H} = \sum_\nu p_\nu \dot{q}_\nu - \mathcal{L}. \tag{2.9}$$

This formalism can be applied to circuits composed by branches with inductors and capacitors. The generalized coordinates are the charges of the capacitors in each branch Q_b, and their corresponding conjugate momenta are the flux of the inductors ϕ_b of the same branch. According to this convention, the kinetic energy of the system is the energy in the inductors and the potential energy is the energy of the capacitors. This leads to the Lagrangian

$$\mathcal{L} = \sum_b \frac{\phi_b^2}{2L_b} - \frac{Q_b^2}{2C_b}, \tag{2.10}$$

and it follows that the Hamiltonian is

$$\mathcal{H} = \sum_b \frac{\phi_b^2}{2L_b} + \frac{Q_b^2}{2C_b}. \tag{2.11}$$

We can make use of the canonical quantization, which in this case simply reduces to replace the flux and charge variables by the quantum operators. Now Q_b and ϕ_b, instead of representing the *values* for the classical charge and magnetic flux in branch b, they represent operators acting on the quantum state of the system. As they are conjugate operators they must obey the commutation relation

$$[\phi_b, Q_b] = \phi_b Q_b - Q_b \phi_b = i\hbar, \tag{2.12}$$

where i denotes the imaginary unit. This equality implies that Q_b and ϕ_b cannot be measured simultaneously. That is, there is no state with fixed charge *and* fixed flux for a branch b. In the following Subsection, we will study the network comprising only one inductor and one capacitor, called LC circuit. There, the commutation relation will be employed to obtain a spectral analysis of the circuit.

2.1.1 LC circuit

We will analyse the example of an LC circuit, i.e. a network comprising a capacitor and an inductor connecting a node with the ground node, as depicted in Fig 2.2. For this particular case, the Hamiltonian in Eq. 2.11 reduces to

$$\mathcal{H} = \frac{\phi^2}{2L} + \frac{Q^2}{2C}. \tag{2.13}$$

As in a classical LC circuit, this Hamiltonian can be related to the one-dimension harmonic oscillator

$$\mathcal{H}_{\text{harm}} = \frac{p^2}{2m} + m\omega^2 \frac{x^2}{2}, \tag{2.14}$$

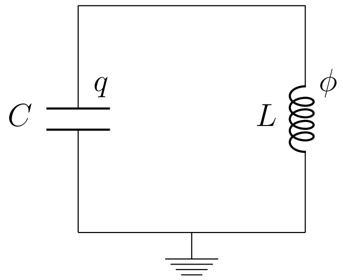


Figure 2.2: Simplest LC circuit.

with the analogy $p \leftrightarrow \phi$, $x \leftrightarrow q$, $m \leftrightarrow C$ and $\omega \leftrightarrow 1/\sqrt{LC}$. Let us now introduce the creation and annihilation operators defined by

$$a = \sqrt{\frac{1}{2\hbar\omega L}}\phi + i\sqrt{\frac{\omega L}{2\hbar}}q, \tag{2.15}$$

$$a^\dagger = \sqrt{\frac{1}{2\hbar\omega L}}\phi - i\sqrt{\frac{\omega L}{2\hbar}}q, \tag{2.16}$$

where i denotes the imaginary unit and † denotes the Hermitian conjugate of the operator. Introducing these operators, the Hamiltonian can be written as

$$\mathcal{H} = \hbar\omega(a^\dagger a + \frac{1}{2}). \tag{2.17}$$

These operators fulfil the commuting property $\left[a, a^\dagger\right] = 1$. It is well known that there exists an eigenbasis $|n\rangle$, with $n = 0, 1, 2...$, in which the action of these operators is

$$a\,|n\rangle = \sqrt{n}\,|n-1\rangle\,, \quad \text{and} \quad a^\dagger\,|n\rangle = \sqrt{n+1}\,|n+1\rangle\,. \tag{2.18}$$

It can be shown that the states $|n\rangle$ are eigenstates of the Hamiltonian with energy $\hbar\omega(n + 1/2)$,

$$\mathcal{H}\,|n\rangle = \hbar\omega(n + \frac{1}{2})\,|n\rangle = \frac{\hbar}{\sqrt{LC}}(n + \frac{1}{2})\,|n\rangle\,. \tag{2.19}$$

This leads to an infinite spectrum of energy levels separated by an energy gap $E_{n+1} - E_n = \hbar\omega$, which is independent of n.

2.2 Josephson junction

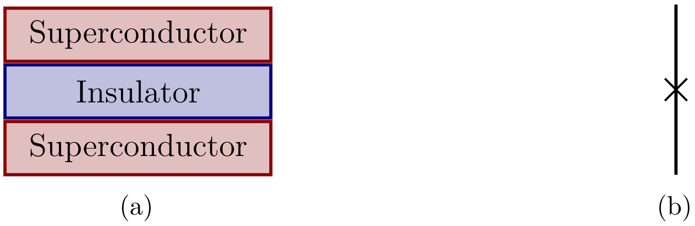


Figure 2.3: Josephson junction, (a) diagram representing the junction and (b) circuit representation for an ideal Josephson junction.

When two superconducting materials are separated by an insulator, a current flux arises due to quantum tunnelling. This structure can be used to obtain a new circuit element known as Josephson junction [11], which is depicted in Fig. 2.3. As two superconducting materials are similar to two parallel metallic plates, Josephson junctions show also a capacitive behaviour besides the superconducting current corresponding to the quantum tunnelling. This two phenomena are usually considered separately, represented the real Josephson junction as an ideal one with a capacitor in parallel.

The ideal Josephson junction can be modelled as a non-linear inductor. The equations relating the current, $i(t)$, through the junction and its voltage drop, $v(t)$, strongly depend on the magnetic flux through the junction, $\Phi(t)$. More precisely, the equations are

$$i(t) = I_c \sin(\varphi(t)) \quad \text{and} \quad v(t) = \frac{\Phi_0}{2\pi}\frac{\mathrm{d}\varphi(t)}{\mathrm{d}t}, \tag{2.20}$$

where I_c is the critical current, Φ_0 denotes the quantum of magnetic flux and $\varphi = 2\pi\Phi/\Phi_0$ denotes a phase representing the reduced flux. The magnetic flux quantum can be calculated from the Planck constant h and the charge of the electron e, $\Phi_0 = h/(2e)$.

Recalling Eq. 2.1, we introduce the equations for the Josephson junction and change the integration variable to obtain the energy of this element

$$\varepsilon_J = \int_{-\infty}^{t} I_C \sin\varphi(t')\frac{\Phi_0}{2\pi}\frac{\mathrm{d}\varphi(t')}{\mathrm{d}t'}\,\mathrm{d}t' = \int_0^{\varphi(t)} I_C \frac{\Phi_0}{2\pi}\sin\varphi\,\mathrm{d}\varphi = E_J(1-\cos\varphi(t)), \tag{2.21}$$

where the constant $E_J = I_C\Phi_0/2\pi$.

2.3 The flux qubit

The fundamental unit in the quantum processor is the qubit, that is a bit with quantum properties, as defined in Sec. 1.1. Now, we will show a method to physically implement a qubit with superconducting circuits, leading to the flux qubit. This qubit is based on the quantization of the magnetic flux, employing the lowest energy levels. There are other superconducting qubits like the charge qubit or the phase qubit, but we will focus on the flux qubit due to its use in adiabatic quantum computing. In particular, the flux qubit has been used to build adiabatic quantum processors by D-Wave [8].

Flux qubits are implemented with rf-SQUIDs, meaning Superconducting QUantum Interference Devices, and RF signals are used to control the magnetic fluxes. The Josephson junctions are the main elements in the rf-SQUID. The simplest configuration is represented in Fig. 2.4a, which is a loop with a Josephson junction and a coil that represents the inductance of the loop. The loop is under the influence of a magnetic flux, Φ^x_{act}, which is used to act on the circuit and tune the value of the phase φ in Eq. 2.20. More complex configurations are also possible, like the compound Josephson junction in Fig. 2.4b, where the smaller loop is biased by a magnetic flux, Φ^x_{co}, which enables more accurate controlling.

For the rf-SQUID with a single junction the phase φ of the junction can be equated

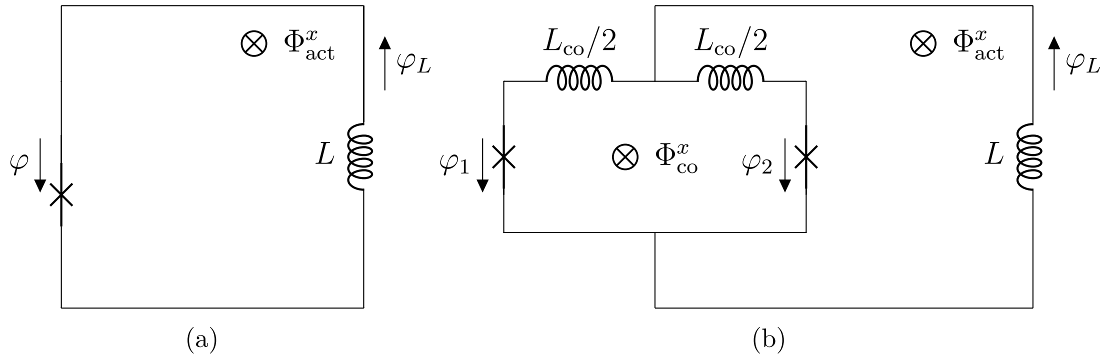


Figure 2.4: Circuit for a rf-SQUID. (a) Single junction rf-SQUID and (b) compound Josephson junction rf-SQUID.

to the phase in the inductance $\varphi = \varphi_L$, and the Hamiltonian can be written as [12]

$$\mathcal{H} = \frac{Q^2}{2C} + V(\varphi), \tag{2.22}$$

$$V(\varphi) = \left(\frac{\Phi_0}{2\pi}\right)^2 \frac{1}{L}\left(\frac{(\varphi - \varphi^x)^2}{2} - \beta \cos\varphi\right), \tag{2.23}$$

$$\beta = \frac{2\pi L I_c}{\Phi_0}, \tag{2.24}$$

where C represents the capacitance of the junction and $\varphi^x = 2\pi\Phi^x_{\text{act}}/\Phi_0$, which is usually biased so that $\varphi^x \approx \pi$. If β is increased, two local minima which are separated by an energy barrier appear in the potential $V(\varphi)$. This leads to two low energy states separated by a big energy gap from the next excited state, typically in the order of $\hbar/\sqrt{LC}$. In the low energy regime, the effective quantum Hamiltonian can be simplified considering only two energy levels as

$$H = -\frac{1}{2}\left(\epsilon\sigma_z + \Delta\sigma_x\right) = -\frac{1}{2}\begin{pmatrix} \epsilon & \Delta \\ \Delta & -\epsilon \end{pmatrix}, \tag{2.25}$$

where ϵ contains the terms biasing the system towards one of those states and Δ represents the terms associated with the energy of the transition from a state to the other. This Hamiltonian is represented in the basis of the persistent currents, which are the eigenstates when the device is strongly biased by Φ^x_{act} ($\epsilon \gg \Delta$), which are counter-circulating persistent current states (clockwise/anticlockwise). The matrices σ_z and σ_x are defined in Eq. 1.4 and Eq. 1.6, respectively. The constants in the effective two-level Hamiltonian can be controlled by external parameters, which makes it possible to use this system as a computational qubit.

2.4 Connecting flux qubits

Several SQUIDs can be connected through magnetic induction [13]. Such a magnetic coupling can be represented by $J_{ij}\sigma_z^{(i)}\sigma_z^{(j)}$, just like for the coupling between spins in Sec. 1.3. In this case, J_{ij} can be a positive or negative constant representing the strength of the coupling, and $\sigma_z^{(i)}$ represents an operation in the state of the ith SQUID.

Combining the individual terms and the coupling terms the two-state Hamiltonian for a single flux qubit becomes a quantum Ising Hamiltonian, as defined in Sec. 1.3,

$$H = -\sum_i \frac{1}{2}\left[\epsilon_i \sigma_z^{(i)} + \Delta_i \sigma_x^{(i)}\right] + \sum_{i<j} J_{ij}\sigma_z^{(i)}\sigma_z^{(j)}. \tag{2.26}$$

This shows that this circuit implementation can be effectively used to implement adiabatic quantum computation.

To perform quantum adiabatic computation, the terms of the Hamiltonian are initially set so that for any i or j we have $\Delta_i \approx \Delta \gg \epsilon_i, J_{ij}$. This leads to $H_0 = -\sum_i \Delta_i \sigma_x^{(i)}/2$ as the effective initial Hamiltonian. As the ground state for each SQUID is a superposition of the two base states (see Eq. 1.9 for the eigenstates of σ_x), the ground state of the whole system is a superposition of all the possible states in the basis. Such an initial state can be easily obtained. Then, the parameters are driven towards the values encoding the problem Hamiltonian, so that effectively

$$H_P = -\sum_i \frac{\epsilon_i}{2}\sigma_z^{(i)} + \sum_{i<j} J_{ij}\sigma_z^{(i)}\sigma_z^{(j)}, \tag{2.27}$$

where ϵ_i represents the individual biases and J_{ij} the couplings for the problem.

2.5 Computational architecture

Ideally a quantum adiabatic processor would allow for couplings between any pair of SQUIDs, that is, J_{ij} could be non-zero for any (i, j) pair, but this is not generally possible. The technical implementation constrains the connectivity between SQUIDs. We will focus on the strategy employed by D-Wave to arrange the connectivity network. The connections of the SQUIDs follow the Chimera architecture. Each SQUID is represented by a node in a Chimera graph, which shows the connections allowed. Recently a new connectivity architecture has been proposed based on Pegasus graph [14].

A Chimera graph is arranged in unit cells, as the one depicted in Fig. 2.5, which are mutually connected. Each cell contains four vertical nodes and four horizontal nodes, each connected to the ones in the perpendicular direction. Each node in the cell is also connected to two nodes in adjacent cells, which are the same position as the first node but in their own cells. The ones in the vertical line are connected with the closest cells in the horizontal direction of the grid and the ones in the horizontal line are connected to the closest cells in the vertical direction of the grid. The connections are also depicted in Fig. 2.5, employing different colours for the connections within the same cell and the connections with adjacent cells.

Usually, we need a particular connectivity different to the one provided by the Chimera architecture. This limitation can be overcome by merging nodes in the graph, forming a chain representing a single node in the original problem. The coupling within a chain has to be considerably stronger than the ones imposed between chains, to make them act as a single node.

Take the example of three connected nodes, as in the two-puppet dilemma example in Sec. 1.3. There are three problem nodes (n_1, n_2 and n_3) and connections between the three of them are required (J_{12}, J_{13} and J_{23}). We can map problem nodes n_1 and

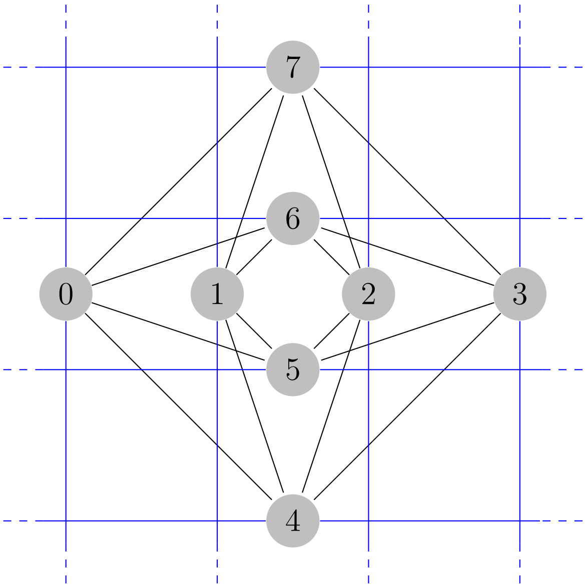


Figure 2.5: Unit cell in a Chimera graph. Black lines represent connections with the perpendicular nodes in the same cell, blue lines represent the connections of each nodes with two nodes in adjacent cells.

n_2 to the graph nodes 1 and 5, while n_3 can be represented by the chain comprising nodes 2 and 6 (see Fig. 2.5). Then, the problem coupling J_{12} would be used in the graph coupling between nodes 1 and 5. The problem couplings J_{13} and J_{23} would be used in the graph couplings between 1 and 6, and 2 and 5, respectively. Finally, an stronger coupling would be imposed between nodes 2 and 6, so that we ensure that they point in the same direction. This configuration is shown in Fig. 2.6. The mapping process that relates an original node distribution to nodes and chains in a Chimera graph is called *minor embedding*.

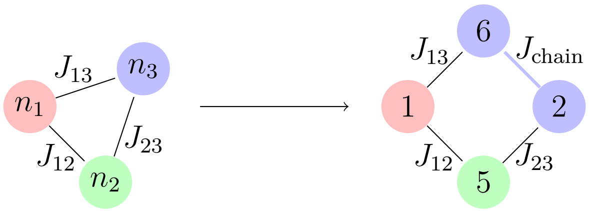


Figure 2.6: Example of minor embedding. The couplings satisfy $|J_{\text{chain}}| \gg |J_{12}|, |J_{13}|, |J_{23}|$.

D-Wave provides tools to embed a problem automatically. Nevertheless, understanding this process is necessary in order to keep in mind the limitations of this technology. For instance, although 16×16 grids with 2048 nodes are available, the effective number of nodes can rapidly decrease when a highly-connected topology is required.

2.6 Overview of the QPU operation cycle

We have so far reviewed the physical implementation in superconducting circuits of a quantum processing unit (QPU). From the fundamental units, implemented with rf-SQUIDs, to their connectivity. Now we will briefly explain how the QPU operates to attend a given request.

The operation cycle of the QPU is divided in two phases: the programming cycle and the anneal-read cycle. During the first phase, the values introduced for a particular problem, represented in H_P, are transformed into the raw signal that will control the magnetic fields of the rf-SQUIDs, shown in Fig. 2.4. This includes performing the mapping of the minor embedding and afterwards performing the transformation to the analog signal. This transformation is performed by the room-temperature electronics performing classical digital-to-analog conversion (DAC). Then, the anneal-read cycle consists in the implementation of the quantum adiabatic evolution procedure. The process is repeated several times, as many times as specified by the user, each final state is measured and returned as a read to the user. Each annealing process takes varying time T according to the annealing time defined by the user. The process is represented in the flow diagram of Fig. 2.7.

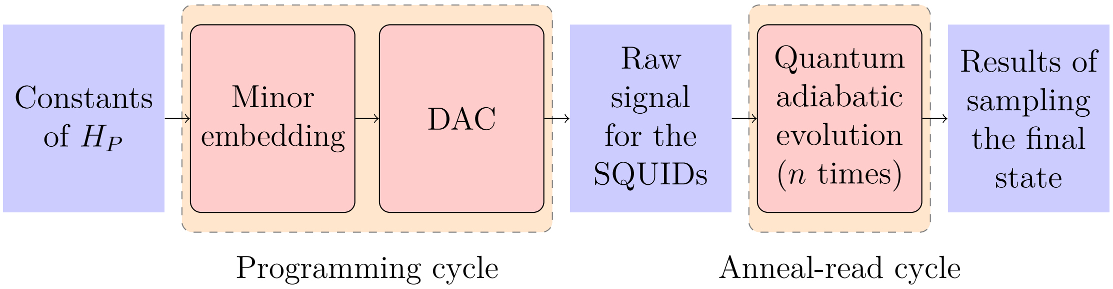

Constants of H_P
Minor embedding
DAC
Raw signal for the SQUIDs
Quantum adiabatic evolution (n times)
Results of sampling the final state
Programming cycle
Anneal-read cycle

Figure 2.7: Flow diagram for QPU operation process.

Chapter 3

Mathematical model of forecasting election polls

This Chapter summarises the main results shown in R. Ibarrondo et al [6], where a spin system has been proposed to model political or poll election and used to forecast the result. Additionally, a practical example employing data from social networks is proposed, with the future goal of running their experiment in an available adiabatic quantum computer.

This Chapter begins with an introduction about the role of physical models in the description and prediction of social behaviour and phenomena. Afterwards, I introduce the concept of political compass, which is a useful representation of the ideology of an individual. In the following section, I describe the political Hamiltonian whose ground state configuration represents the expected outcome of election polls. This leads to a review of the experimental implementation and the data retrieval from *Twitter* to set the free parameters. Finally, I describe how this model could be solved in an adiabatic quantum computer.

3.1 Physical models to describe social behaviour

Mathematical models inspired in physical systems have been widely employed to describe the collective social behaviour [2]. For example, molecular dynamics in gases inspired a model to describe the collective behaviour of assistants in heavy metal concerts [4]. People were represented as particles in a gas, whose collective motion was simulated and their analysis of available videos of heavy metal concerts showed a strong accordance with their predictions.

We have already described that some problems can be codified in the Ising Hamiltonian given in Eq. 1.28. In particular, we are specially interested in applying it to the study of social dynamics. For example, in Ref. [3] an Ising Hamiltonian was used to study social opinion interaction within neighbourhoods where neighbouring individuals tend to think alike. They represented a city by a spin grid, that is, a classical spin was assigned to each point in the grid, $s_i = \pm 1$, involving that neighbouring spins tend to align. Due to the spin representation, the tendency to align was straightforwardly introduced with ferromagnetic couplings between neighbouring spins. This couplings were introduced into an Ising Hamiltonian which provides a measure of the energy of the lattice (unhappiness of the neighbourhood), which is to be minimised, and it is given by

$$H = -\sum_{(i,j)} s_i s_j, \tag{3.1}$$

where s_i is the ith spin in the grid and the sum is performed over all neighbouring positions (i, j). If two neighbouring spins happen to have the same direction, $s_i = s_j = \pm 1$, then their contribution to energy is negative $-s_i \cdot s_j = -1$, reducing the energy. While if they are counteraligned, $s_i = -s_j = \pm 1$, their contribution to energy is positive $-s_i \cdot s_j = 1$, increasing the energy. This model was studied generating initial patterns which were evolved iteratively lowering the energy of the system. Finally, the patterns in the final lattices were analysed, which showed a tendency towards clustering. This model reproduces the clustering of ideologies observed in the vote distribution among electoral districts.

These are only a couple of examples modelling social behaviour by physical interactions. Both models have been contrasted with experimental data, showing that their results successfully replicate real social patterns.

3.2 Political compass

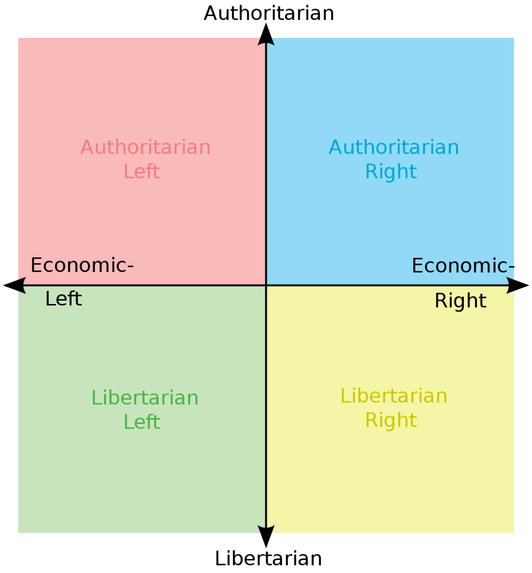


Figure 3.1: Example of a two-axis political compass.

We are familiar with the left-right classification of political parties in a political spectrum, since it simplifies the prediction of possible policies. This is the reason why when a new party emerges, social media quickly try to allocate it in that spectrum. We are also familiar with expressions like "that decision of the party reveals it is turning left/right" to describe the change in the political strategy of a party. To sum up, we are used to describing some aspects of the ideology of a party by using a single axis (left - right), but this is sometimes too simplistic. Indeed, it is reasonable to think that by introducing multiple axes, we can improve the accuracy in the description of the ideology of a party. Typically, when several axes are employed, the term political spectrum is replaced by the more general concept of political compass. Ideally, a political compass has a range of different axes that altogether can precisely describe the ideology of either a political party or a citizen. Roughly speaking, in the political context, two main axes would be the economic axis "left - right" and the social axis "authoritarian - libertarian". In Fig. 3.1, a representation of the space of possibilities with two-axis is shown. For a more detailed discussion about the topic see Ref. [5].

The political compass is more than a theoretical concept. In fact, knowing the ideology of an individual provides valuable information about what it is willing to vote. Particularly, we can assume that its decision will tend towards the position in the political compass space. Altogether, if the position of each individual in the political compass is known beforehand, predicting the outcome of a voting process is straightforward. The model described in this Chapter deals with the estimation of the position of each individual in the political compass in order to predict the outcome of referendums or election polls.

Determining the vector corresponding to each individual is undoubtedly not an easy task. In the following section, we will consider several aspects which affect the position of the political compass. First, we will consider the initial ideological background of each individual. We will also introduce the impact of external events, such as crisis scenarios or controversial acts of political entities. And finally, we will consider the vast amount of social interactions between individuals themselves, which mutually modify their position in the political compass.

3.3 Political Hamiltonian

In this Section, I will review the model introduced in Ref. [6] and I will focus on the case of a single-axis political compass. The position in an axis is described by a vector in a two dimensional space, $\vec{S}_i$, which represents a classical spin. More explicitly, it is determined by the projection of the vector $\vec{S}_i$ in it, $\cos\theta_i$, which is computed with the angle $\theta_i \in [0, 2\pi)$. For the limit cases where the spin is completely aligned in the axis ($\theta_i = 0$ or π) we have that the projection is either $\cos\theta_i = 1$ or -1.

Representing the position in the political compass with a spin vector allows us to find an Ising Hamiltonian that we call political Hamiltonian. As I explained in Section 1.3, biases can be used to consider preferences and couplings between spins can be used to consider interactions between individuals. In the following we will describe how to build the political Hamiltonian, whose ground state configuration is considered to be the expected configuration.

First, we will introduce opinion preferences of each individual, $\vec{B}_i^{\text{pref}}$, which represent the ideological background of each individual. We should also consider, in the same manner, the effect of external agents as individual magnetic fields, $\vec{B}_i^{\text{ext}}$. This agents could be either social events which can affect public opinion or even the influence of a trending topic video. In this case, although the external agent may be the same for every person, individual magnetic fields are considered because the effect on each person may vary strongly. Both magnetic fields can be interpreted as components of a single individual bias $\vec{B}_i = \vec{B}_i^{\text{pref}} + \vec{B}_i^{\text{ext}}$. The effect of the individual biases is introduced in the Hamiltonian as the term

$$-\sum_i \vec{B}_i \cdot \vec{S}_i. \tag{3.2}$$

We must also consider the interaction between individuals, following a similar reasoning to the deduction of Eq. 3.1,

$$-\sum_{i,j} J_{ij} \vec{S}_i \cdot \vec{S}_j, \tag{3.3}$$

where J_{ij} are the coupling constants. When the interaction is ferromagnetic, $J_{ij} > 0$, the spins tend to be aligned. While if the interaction is antiferromagnetic, $J_{ij} < 0$, coupled spins tend to point in opposite directions. The ferromagnetic (antiferromagnetic) interaction is applied between individuals which tend to think alike (differently). The strength of the interaction is represented by the absolute value of the coupling constant, $|J_{ij}|$. This allows us to distinguish strongly interacting individuals from the weakly interacting ones. In fact, for a non-interacting pair we simply consider $J_{ij} = 0$.

Under the consideration of the ideological background ($-\vec{B}_i^{\text{pref}} \cdot \vec{S}_i$), the effects of external events ($-\vec{B}_i^{\text{ext}} \cdot \vec{S}_i$), and the pairwise interaction ($-J_{ij}\vec{S}_i \cdot \vec{S}_j$), we can define the political Hamiltonian as

$$H = -\sum_i \vec{B}_i \cdot \vec{S}_i - \sum_{i<j} J_{ij} \vec{S}_i \cdot \vec{S}_j, \tag{3.4}$$

where the second sum is performed for every different i, j pair. Consequently, by finding the ground state configuration for the whole system, we can obtain the the spin state $\vec{S}_i$ for each individual. Knowing those states will allow us to perform a prediction.

Precisely, if we focus on the case of two-option referendums the political Hamiltonian has the form

$$H = -\sum_i B_i s_i - \sum_{i<j} J_{ij} s_i s_j, \tag{3.5}$$

where we replaced $\vec{S}_i$ by $s_i = \pm 1$, as we only care about the extreme cases in the axis. According to Section 1.3 this Ising Hamiltonian can be described in terms of quantum spins as

$$H = -\sum_i B_i \sigma_z^{(i)} - \sum_{i<j} J_{ij} \sigma_z^{(i)} \sigma_z^{(j)}, \tag{3.6}$$

with $\sigma_z^{(i)}$ representing the z-direction spin operator, which makes it suitable for quantum adiabatic computation.

It is worthy to mention a previous work employing similar Hamiltonians to reproduce human opinion interaction and applied to forecasting election outcomes [15]. Nevertheless, the focus on previous research is done in the analysis of the dynamics. In this case, instead of looking for a dynamical analysis, the aim is to obtain the ground state that would represent the most reasonable output.

3.4 Experimental procedure

We have tested the model introduced in the previous Section by running a simple experiment employing data from social networks [6]. Indeed, using a small group of volunteers, who allowed us to collect data from their social networks, more precisely, from *Twitter*, to construct the model and, afterwards, they made a two option poll whose outcome we tried to predict using our model. In order to construct the political Hamiltonian, we used their latest 200 tweets and latest 200 likes. The interactions were restricted to positive interactions which are considered in a number a_{ij} that represents the amount of positive reactions which individual j gave to i. Then, these constants are used to obtain estimations for the parameters in Eq. 3.4.

As the coefficients a_{ij} are not symmetric the coupling between individuals is defined as

$$J_{ij} \equiv \frac{1}{2}(a_{ij} + a_{ji}). \tag{3.7}$$

The individual fields, h_i, which play the role of $\vec{B}_i$ in Eq. 3.4, were estimated considering the total amount of positive reactions one individual produces,

$$h_i \equiv f_i \sum_j a_{ij}, \tag{3.8}$$

with the constant f_i reflecting the initial preferences of the individual i. It will be set to ± 1 if the preference of that individual is known and to 0 if no previous preference is determined.

Recalling that we will consider decisions with only two possible choices the spin Hamiltonian reads

$$H = \sum_i h_i s_i - \sum_{i,j} J_{ij} s_i s_j. \tag{3.9}$$

More specifically, our experiment involved 10 individuals whose interactions were employed to obtain the parameters for the Hamiltonian. Once the couplings were obtained a survey with 9 questions was elaborated. This questions were of the form choose "A" or "B", and stand for possible referendum options. Although the coupling factors are assumed to be the same for every question, the initial preferences, f_i, must be adjusted depending on the question. Nevertheless, we do not have a measure of the biases before the survey is done, thus we will use part of the information obtained from the outcome of the survey in order to estimate their preferences. Precisely, we will consider neutral preferences for most of the individuals, $f_i = 0$, while fixing it, $f_i = \pm 1$, for a small subset by directly checking their answers to the survey.

In the following Section, I review the procedure to obtain the data from *Twitter* for setting the parameters of the Hamiltonian in the aforementioned experiment. Let us now advance the results from those interactions, which are condensed in the matrix a, which reads

$$a = \begin{pmatrix} 0 & 3 & 2 & 1 & 4 & 11 & 21 & 3 & 0 & 2 \\ 2 & 0 & 3 & 3 & 1 & 2 & 2 & 0 & 0 & 0 \\ 1 & 3 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 7 & 6 & 1 & 1 & 0 & 5 & 12 & 56 & 0 & 6 \\ 1 & 1 & 0 & 1 & 2 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 2 & 1 & 2 & 1 & 2 & 1 & 0 & 0 \end{pmatrix}. \tag{3.10}$$

Employing this data we are able to construct the coupling matrix, J, given by Eq. 3.7

$$J = \begin{pmatrix} 0.0 & 2.5 & 1.5 & 0.5 & 5.5 & 6.0 & 11.0 & 2.0 & 0.0 & 1.5 \\ 2.5 & 0.0 & 3.0 & 2.0 & 3.5 & 1.5 & 1.5 & 0.0 & 0.0 & 1.0 \\ 1.5 & 3.0 & 0.0 & 0.5 & 0.5 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \\ 0.5 & 2.0 & 0.5 & 0.0 & 0.5 & 1.0 & 0.5 & 0.0 & 1.0 & 0.5 \\ 5.5 & 3.5 & 0.5 & 0.5 & 0.0 & 3.5 & 6.5 & 28.5 & 0.0 & 4.0 \\ 6.0 & 1.5 & 0.0 & 1.0 & 3.5 & 0.0 & 1.0 & 0.0 & 1.0 & 1.0 \\ 11.0 & 1.5 & 0.0 & 0.5 & 6.5 & 1.0 & 0.0 & 0.5 & 0.5 & 1.5 \\ 2.0 & 0.0 & 0.0 & 0.0 & 28.5 & 0.0 & 0.5 & 0.0 & 0.0 & 1.0 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.0 & 1.0 & 0.5 & 0.0 & 0.0 & 0.0 \\ 1.5 & 1.0 & 1.0 & 0.5 & 4.0 & 1.0 & 1.5 & 1.0 & 0.0 & 0.0 \end{pmatrix}. \tag{3.11}$$

Although the individual fields, h_i, depend on each question let us show the case where all the initial preferences are set aligned, $f_i = 1$,

$$h^* = \begin{pmatrix} 47 & 13 & 5 & 3 & 94 & 8 & 8 & 3 & 2 & 12 \end{pmatrix}, \tag{3.12}$$

which shows the maximum absolute value each h_i can have.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
A	-1	$+1$	$+1$	-1	-1	$+1$	-1	-1	$+1$
B	-1	$+1$	-1	-1	$+1$	-1	$+1$	-1	-1
C	$+1$	-1	$+1$	$+1$	-1	$+1$	-1	$+1$	$+1$
D	$+1$	-1	$+1$	$+1$	-1	$+1$	$+1$	$+1$	$+1$
E	$+1$	$+1$	$+1$	-1	-1	$+1$	-1	-1	$+1$
F	$+1$	-1	-1	$+1$	$+1$	-1	$+1$	$+1$	-1
G	$+1$	$+1$	-1	$+1$	$+1$	-1	$+1$	$+1$	-1
H	-1	-1	-1	$+1$	$+1$	-1	$+1$	$+1$	-1
I	-1	$+1$	$+1$	-1	-1	$+1$	-1	-1	$+1$
J	-1	$+1$	-1	$+1$	$+1$	-1	$+1$	$+1$	$+1$
Score	50%	50%	67%	62.5%	56%	89%	62.5%	62.5%	75%
Fixed	E, G	A, C	A	C, G	G	A	C, D	C, G	A, G

Table 3.1: Results obtained from the survey and the accordance with our prediction. For the 9 questions (Q1 - Q9) the answer given by each individual (A - J) is shown, together with the prediction score obtained by the model. We also show which individuals were fixed for each question.

In Table 3.1 we show the answer each individual (A - J) gave to each question (Q1 - Q9). We also show the proportion of correctly predicted answers, which gives a score of the correctness. According to the calculations performed, the model reproduces some of the scenarios with great accuracy. We can check that the score obtained is above the 50% in all of them and reaches even $70 - 90\%$ for some questions.

3.5 Gathering the data from *Twitter*

For the purpose of collecting interactions in the social network, the *Twitter* API was accessed by means of the `Twython` library available for Python. First, we will explain how the test group was selected and then the data obtained from their interactions. For a brief description of the *Twitter* API, see Appendix A. The Python programs I developed to perform the tasks described in this Section are available in the repository of the University of the Basque Country (ADDI).

3.5.1 Deciding the test group

In order to have a suitable test group for the experiment, our aim was to find at least 10 highly interacting individuals. Those users were chosen from the scientific community close to quantum technologies, which enabled topic related questions and facilitated the obtention of their permission. The challenge was to find a small group that was also highly interacting in *Twitter*.

Initially 5 users were selected based on a qualitative insight. Once their mutual interaction was ensured, we faced the task of getting at least 15 additional candidates. The reason is to account with a margin in case some of them refuse to take part in the survey. We tried to get the remaining participants from the *Twitter* contacts of one of the members of the community, also connected with the 5 members of the initial group. This provided us with near to 400 candidates.

The main objective is to find the most interacting subgroup from the 400 candidates, which included the 5 initial users. In order to construct an optimal solution, we designed a sorting criteria based on the interactions of the individuals with the initial group. Moreover, once the interactions between users are measured, it is straightforward to find the individuals that interact the most with the initial ones, but we also want those that mutually interact strongly. Therefore, we designed a net analysis inspired by eigenvector centrality and PageRank, an algorithm used to rank webpages [16].

Firstly, the interactions in the complete network of candidates were characterized and the initial group (seed-group) was defined. Our ranking algorithm worked as follows,

1. Initially each individual in the net has 0 score points.
2. An initial score point, s_o, is added to each individual in the seed-group.
3. Each individual distributes their score, dealing to each target individual a portion proportional to the connection the source individual has with the target individual. The score shared is computed by dividing by a decrease factor, d, the original score of the individual.
4. Each individual in the seed-group gets an additional score increment, s_i.
5. Repeat steps 3 and 4 until an ending criteria is met.

The initial score and the score increment of the seed-group ensures they behave as the 'score-source' in the score sharing process. Similarly, applying a decrease factor prevents from selecting highly interacting users who are too far from the seed-group.

For this case, the choice $s_o = 1$, $d = 1.05$ and $s_i = 0.1$ perfectly worked iterating 100 times. We performed this twice, one implementing connections as follow/unfollow relations and other considering the likes they gave to each other. From the best ranked individuals, we selected the candidates for the experiment and asked for their permission to take part in the experiment.

3.5.2 Analysing the interactions

We will now consider the ten individuals named from A to J, in order to analyse their interactions. According to the restrictions of *Twitter* API, we were only allowed to employ data from the latest 200 posted and liked tweets. Depending on the user, this usually varies from 1 to 4 months.

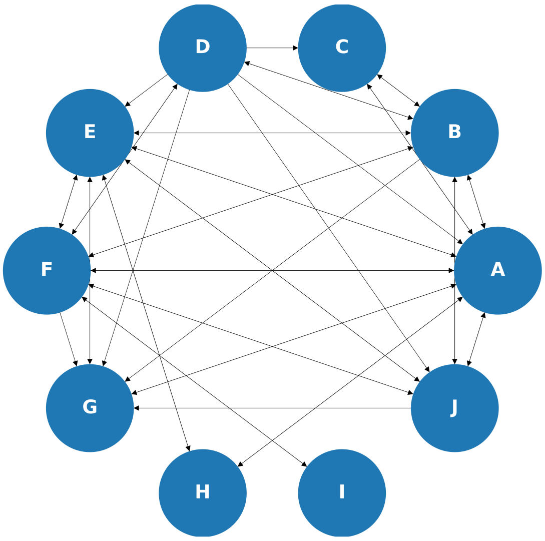


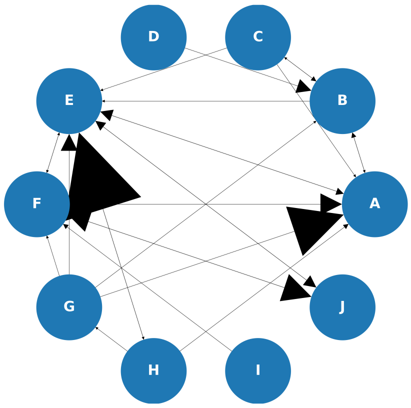
Figure 3.2: The interactions between users based on follows. The arrows show which users are followed by a given user.

The first part of the analysis consists in checking whether the users mutually follow. In Fig. 3.2, we show a directed graph in which each arrow represents connected individuals. The arrow indicates who follows whom, and a double arrow means that they mutually follow. This is useful to check that the selected group is highly connected.

The arrows depicted in Fig. 3.3a represent the amount of times an individual liked a tweet posted by other individual. The data depicted in this plot is provided in Table 3.3b.

Finally, arrows in Fig. 3.4a represent the amount of times a user retweeted other. The data depicted in this graph is given in Table 3.4b.

We interpreted this interactions as positive interactions, reflecting the strength of a ferromagnetic coupling between each pair of individuals. Finding a negative coupling may be trickier, as *Twitter* does not provide a dislike button. It would have been interesting to analyse the language used in the retweeted or replied tweets, searching for negatively intended messages. Nevertheless, finding a reliable method to perform that measurement without a great risk of false positives would be tough and out of the scope of this project. It is worth, nonetheless, to explore this path if this model is applied to a more elaborated decision process.

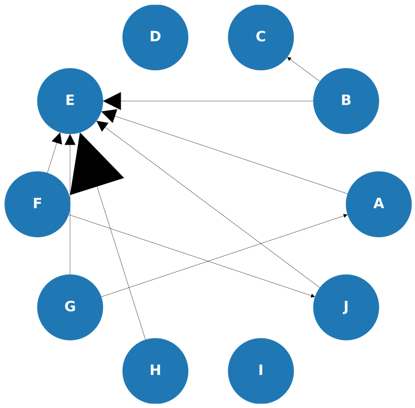


(a)

user 1	user 2	1 liked 2	2 liked 1
A	B	2	1
A	C	0	1
A	E	5	3
A	F	0	9
A	G	0	22
A	H	0	2
B	C	1	2
B	D	0	6
B	E	1	0
B	G	0	1
C	E	1	0
E	F	1	1
E	G	0	7
E	H	1	34
E	J	5	4
F	G	0	1
F	I	0	2
F	J	12	9
G	H	0	1

(b)

Figure 3.3: The interactions between users based on likes, (a) graph representation of the interactions and (b) table showing the raw data.



user 1	user 2	1 RTed 2	2 RTed 1
A	E	4	0
A	G	0	1
B	C	1	0
B	E	5	0
E	F	0	3
E	G	0	3
E	H	0	16
E	J	0	3
F	J	1	0

(a) (b)

Figure 3.4: The interactions between users based on retweets (RTs), (a) graph representation of the interactions and (b) table showing the raw data.

3.6 Forecasting with quantum adiabatic computers

For the particular case of 10 individuals, the model can be computed with classical resources, which is what was attained in Ref. [6] as a proof of principle. Nevertheless, it is also possible to solve the model using quantum adiabatic computation and it is meaningful for larger networks. In this Section, I describe a simulated quantum adiabatic computation process using the software offered by D-Wave [8]. The simulation of this example shows that the parameters of the model can be easily codified as a natural input for quantum adiabatic computers.

I employed the `SimulatedAnnealingSampler` class, available in the `neal` library offered by D-Wave. With this class we can simulate a request to a quantum adiabatic computer, using the very same input and output formats. In order to use the method `sample_ising`, which reproduces the results we would obtain, we need to recast h_i and J_{ij} to a Python dictionary format. For example,

```
h = {0: 0, 1: 0, 2: −5, 3: 0, 4: 0, 5: 0, 6: 8, 7: 0,                     1
              8: 0, 9: 0}                                                   2
J = {(0, 1): −2.5, (0, 2): −1.5, (0, 3): −0.5, (0, 4): −5.5,               3
        (0, 5): −6.0, (0, 6): −11.0, (0, 7): −2.0, (0, 9): −1.5,            4
        (1, 2): −3.0, (1, 3): −2.0, (1, 4): −3.5, (1, 5): −1.5,             5
        (1, 6): −1.5, (1, 9): −1.0, (2, 3): −0.5, (2, 4): −0.5,             6
        (2, 9): −1.0, (3, 4): −0.5, (3, 5): −1.0, (3, 6): −0.5,             7
        (3, 8): −1.0, (3, 9): −0.5, (4, 5): −3.5, (4, 6): −6.5,             8
        (4, 7): −28.5, (4, 9): −4.0, (5, 6): −1.0, (5, 8): −1.0,            9
        (5, 9): −1.0, (6, 7): −0.5, (6, 8): −0.5, (6, 9): −1.5,             10
        (7, 9): −1.0}                                                       11
```

To determine the directions of h_2 and h_6 their answers were checked, as explained in Section 3.4. Then the quantum adiabatic process is simply simulated with the following lines:

```
solver = neal.SimulatedAnnealingSampler()                                  12
sampleset = solver.sample_ising(h, J, num_reads=10)                        13
output = sampleset.first[0]                                                14
```

Let me briefly describe this code. In line 12, I create an object which simulates the streamline with a real quantum computer. In line 13, I call a method simulating a request for an experiment with the defined bias and coupling values, which is iterated 10 times. In the `sampleset` object, the information about the different runs of the experiment would be saved, and in line 14, the configuration for the configuration with the lowest energy obtained is accessed and saved in the `output` variable. Finally, we can see the comparative between the prediction and the result obtained in the poll:

$$\begin{aligned} \text{Prediction:} &\quad [-1,\ -1,\ -1,\ -1-1,\ -1,\ -1,\ -1,\ -1,\ -1] \\ \text{Poll:} &\quad [\ 1,\ -1,\ 1,\ -1,\ 1,\ -1,\ -1,\ -1,\ 1,\ -1], \end{aligned}$$

which results in an accordance score of 60% between the prediction and the poll.

Conclusions

Social dynamics have been successfully described employing mathematical models inspired by physical systems. Along these lines, we have described a model that can forecast the results of two option referenda, codifying it in a spin Hamiltonian representing a network of individuals. Additionally, we have shown how this problem can be solved using quantum adiabatic computation, which is a computational paradigm that uses the principles of quantum adiabatic evolution to find configurations minimising given cost functions.

In the first chapter, an introduction to quantum adiabatic computation is provided starting from the most fundamental concepts of quantum mechanics and understanding the representation of the Hamiltonian in quantum systems, which plays the roll of their energy. Due to the adiabatic theorem, a quantum system in its ground state (the configuration that minimises its energy) can be slowly varied towards the final Hamiltonian while keeping it in the ground state. This enables us to find the ground state configuration of an arbitrary Hamiltonian, which can encode the formulation of satisfiability problem, for example.

After introducing the fundamental theoretical concepts of adiabatic quantum computation, we have reviewed the physical implementation of quantum adiabatic computation with superconducting circuits in Chapter 2. This chapter provides a brief description of the technical details of quantum adiabatic processors, which allows us to describe the scope of problems that can be implemented in these platforms. Satisfiability problems can be solved and the minimal energy configuration for Ising Hamiltonians, e.g. the one considered in the final chapter, can be found as far as the size of the network and its connectivity is limited.

Finally, the last chapter is dedicated to describe the model for election forecasting based on spin systems published in Ref. [6]. First, I introduce the mathematical formulation of the model. Afterwards, I show how the data retrieval from *Twitter* was performed, and how the couplings are constructed by means of likes and retweets of individuals within a chosen network. The Python programs I developed to perform this task are available in the repository of the University of the Basque Country (ADDI). Finally, this model is used to predict the possible outcomes of a poll with two option questions performed to the network of individuals.

Altogether, this work shows that quantum adiabatic computation can be used to find optimal configurations for a spin model. In particular, we have applied it to predict the results of polls, which could be used to predict elections and referendums. Additionally, we have reviewed the technical resources provided by D-Wave to implement quantum adiabatic computation, which could be used to solve the political forecasting model.

Future perspectives would include to implement the problem in a D-Wave computer and to use other sources to set the parameters of the model, such as other social networks or preliminary surveys. This could allow us to design a better real interactions in the Hamiltonian. Certainly, testing the model with a larger network would be enriching. This would provide not only a more accurate result from the statistical point of view, but it would also provide the opportunity to perform a deeper analysis of the interactions.

Bibliography

[1] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, "Quantum Computation by Adiabatic Evolution," 2000. [Online]. Available: http://arxiv.org/abs/quant-ph/0001106

[2] A. Basinski-Ferris, "Comparison of Mathematical Models of Opinion Dynamics," *The iScientist*, vol. 2, no. 1, pp. 7–14, 2017. [Online]. Available: https://journals.mcmaster.ca/iScientist/article/view/1357/1170

[3] C. Castellano, S. Fortunato, and V. Loreto, "Statistical physics of social dynamics," *Reviews of Modern Physics*, vol. 81, no. 2, pp. 591–646, 2009.

[4] J. L. Silverberg, M. Bierbaum, J. P. Sethna, and I. Cohen, "Collective Motion of Moshers at Heavy Metal Concerts," 2013. [Online]. Available: http://arxiv.org/abs/1302.1886{%}0Ahttp://dx.doi.org/10.1103/PhysRevLett.110.228701

[5] A. Petrik, "Core Concept "Political Compass" How Kitschelt's Model of Liberal, Socialist, Libertarian and Conservative Orientations Can Fill the Ideology Gap in Civic Education," vol. 9, no. 4, pp. 45–62, 2010.

[6] R. Ibarrondo, M. Sanz, and R. Orús, "Forecasting Election Polls with Spin Systems," pp. 1–7, 2020. [Online]. Available: http://arxiv.org/abs/2007.05070

[7] N. Wiebe and N. S. Babcock, "Improved error-scaling for adiabatic quantum evolutions," *New Journal of Physics*, vol. 14, 2012.

[8] "D-wave main page," https://www.dwavesys.com, accesed: 2020-05-25.

[9] *Technical Description of the D-Wave Quantum Processing Unit*, D-Wave, 2020.

[10] J. A. Russer and P. Russer, "Lagrangian and hamiltonian formulations for classical and quantum circuits," *IFAC Proceedings Volumes*, vol. 45, no. 2, pp. 439 – 444, 2012, 7th Vienna International Conference on Mathematical Modelling. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667016307108

[11] K. K. Likharev and J. Lukens, *Dynamics of Josephson Junctions and Circuits*, 1988, vol. 41, no. 11.

[12] R. Harris, J. Johansson, A. J. Berkley, M. W. Johnson, T. Lanting, S. Han, P. Bunyk, E. Ladizinsky, T. Oh, I. Perminov, E. Tolkacheva, S. Uchaikin, E. M. Chapple, C. Enderud, C. Rich, M. Thom, J. Wang, B. Wilson, and G. Rose, "Experimental demonstration of a robust and scalable flux qubit," *Physical Review B - Condensed Matter and Materials Physics*, vol. 81, no. 13, pp. 1–20, 2010.

[13] R. Harris, T. Lanting, A. J. Berkley, J. Johansson, M. W. Johnson, P. Bunyk, E. Ladizinsky, N. Ladizinsky, T. Oh, and S. Han, "Compound Josephson-junction coupler for flux qubits with minimal crosstalk," *Physical Review B - Condensed Matter and Materials Physics*, vol. 80, no. 5, pp. 1–5, 2009.

[14] N. Dattani, S. Szalay, and N. Chancellor, "Pegasus: The second connectivity graph for large-scale quantum annealing hardware," 2019. [Online]. Available: http://arxiv.org/abs/1901.07636

[15] S. Galam, *Sociophysics: A Physicist's Modeling of Psycho-political Phenomena*. New York: Springer-Verlag, 2012.

[16] J. Grus, *Data Science from Scratch: First Principles with Python*. Beijing: O'Reilly, 2015. [Online]. Available: http://my.safaribooksonline.com/97814919-01427

[17] "Docs - twitter developers," https://developer.twitter.com/en/docs, accesed: 2020-05-25.

Appendix A

Requests to the Twitter API

Here I briefly expose the request types employed, in order to clearly expose the available data and the challenge in its entirety. This information is a summary of the Twitter API Documentation [17] sticking to the relevant information for our purposes.

It is a commonplace to use `user` objects to handle the data about a user. The relevant attributes held within this objects usually are: the user id, the user name, the screen name, the count of followers and the count of friends. They also contain a status object containing creation data, biography text... To identify users within the net we will employ their user ids and map them to a new code when shown, so as to ensure anonymity.

GET friends/ids, GET friends/list. These requests allow us to get the list of users followed by a given user, also called friends. The first one returns only the `ids` of each friend, while the second one provides a `user` object containing complementary information. The first method returns a maximum of 5,000 ides per distinct request, the second a maximum of 200 users. Both allow at most 15 request per 15 minutes. The methods also provide complementary data and parameters to adjust to the requirements of the application.

GET friendship/show. Given a source user and a target user a response will be provided with a `relationship` object. The attributes of `relationship` provide if they follow each other, among others. We are allowed 180 request per 15 minutes.

GET statuses/user_timeline. Returns the latest tweets posted by a given user. This also contains retweeted tweets and some optional parameters are allowed to filter the obtained result. There is a 1,500 request per 15 minutes limit and a maximum of 100,000 request a day. Each request contains a maximum of 200 tweets. Each `tweet` contains its creation data, the user that posted it, its text, if it was in reply to any other tweet or user, the count of retweets and favourites it has. When retweeted you get the whole `user` object of the retweeted account. This is the best way we found to check if users retweeted each other.

GET favourite/list. Returns the list of the latter liked tweets by a specified user. The user that posted the tweet id within the `tweet` object, as mentioned above. A maximum of 200 tweets can be obtained in each request and the request limit is of 75 request per 15 minutes. We are kept from accessing tweets that are too old, limiting our capacity to gather liked tweets.