Master Thesis

Universitat de Barcelona

Master in Quantum Science and Technology

Embedding strategies for adiabatic quantum computation

Supervisor:
Marta P Estarellas

Author:
Juan Alberto Cereijo Freire

Co-supervisors:
Matthias Werner
Ana Palacios
Jordi Riu

29th August 2022

Universitat de Barcelona

Abstract

Quantum Annealing (QA) is an alternative to gate based Quantum Computation (QC) to solve problems not efficiently tractable on classical devices. Right now, QA is advantageous over QC in the noisy intermediate-scale quantum (NISQ) era for its lesser need of error correction codes and the resource overhead they suppose. However, hardware limitations in terms of connectivity and feasible interactions create incompatibilities between the chip and the structure of the problem, which leads to what is called the graph embedding problem. To circumvent this obstacle, we first analyse the current solutions based on heuristic algorithms and their limitations. We then explore the potential of a digital assisted annealing (DaA) approach. The novelty of this technique relies on the fact that the state generated by the quantum annealer is used as the initial state of the variational circuit, the role of which is to approach a target solution the annealer could not reach by itself due to its hardware limitations. We complete this thesis with a detailed study on the performance of our approach for different scenarios and branches we would like to explore.

Acknowledgments

First of all I would like to thank Dr. Marta P Estarellas for supervising this master thesis and for all her work during these months in relation to our stay in Qilimanjaro which were very relevant for me as a learning experience that was not restricted only to the topic covered here.

I would also like to thank my co-supervisors Ana Palacios and Matthias Werner for all the support and guidance they gave me through every stage of this project. Then also to Javier Sabariego for helping me when I needed help with the software tools.

And in general to the warm welcome I received at Qilimajaro where it was a pleasure to spend time with them and they made the breaks and lunch time much more enjoyable.

And last but not least, to my parents for putting up with me all these efforts so that I could pursue this academic career in the most comfortable and carefree way possible.

Contents

1 Introduction

Quantum computation (QC) harnesses quantum mechanical effects such as superposition, interference, and entanglement, to bring potential advantages over classical computing (CC) [1]. The reason QC was proposed as an alternative to CC was to solve problems out of reach for CC [2]. Among others, some are based on combinatorial optimisation or satisfiability, such as travelling salesman or maximum independent set, while others relate to the direct simulation of quantum systems, such as the electronic structure of molecules for chemical simulations. To realise QC, different computing paradigms are being developed. One of the most popular ones is the circuit or gate model, a digital paradigm in which the problem is encoded into a sequence of quantum gates. Its algorithms aim to create the circuits that are applied to an initial easy-to-prepare state to yield a final state encoding the solution. It is an universal QC paradigm, meaning that it can encode any arbitrary quantum problem. Then we have quantum annealing (QA) and adiabatic quantum computation (AQC), where the problem is encoded into the ground state of a final Hamiltonian. Both are analogue and universal, as long as you can construct arbitrary Hamiltonians [3], and while AQC is constrained to adiabatic evolution, QA also allows for diabatic transitions. However, currently available QA and AQC implementations only encode a restricted type of Hamiltonian, thus they are not universal. All these paradigms are in current development and their success relies on the improvement of the hardware. In the AQC framework, we are seeking an approach that is not restricted to the classical Ising model, which is the typical limitation, but that also works well with quantum models. Moreover, between gate QC and AQC, the latter is expected to be more advantageous in the Noisy Intermediate-Scale Quantum (NISQ) [4] era, where only moderately sized and noisy devices are available, for being more robust against errors [5]. We are still quite far from reaching the fault-tolerance quantum computing era for which we will have enough quantum resources and control to correct errors from digital quantum circuits, and here we choose to explore the AQC/QA paradigm instead, which from now on we will generally refer to as quantum annealing (QA).

1.1 Adiabatic model of computation

The AQC model [6] uses the adiabatic theorem [7] to find the global minimum of a function by taking advantage of quantum fluctuations to avoid getting stuck on local minima. An algorithm is constructed by taking an easy-to-prepare initial Hamiltonian, $\mathcal{H}_{init}$, and a desired final Hamiltonian, $\mathcal{H}_{final}$, encoding the problem solution in the ground state of the latter. The system is evolved in a slow manner by interpolating between $\mathcal{H}_{init}$ and $\mathcal{H}_{final}$ such that for $t \in [0, T]$, with T being the total running time, the system's Hamiltonian takes the following form:

$$\mathcal{H}(t) = (1 - s(t))\,\mathcal{H}_{\text{init}} + s(t)\,\mathcal{H}_{\text{final}} \tag{1}$$

where the schedule, s, increases monotonically from $s(0) = 0$ to $s(T) = 1$ alongside the evolution path: $s : [0; T] \Rightarrow [0; 1]$. Time complexity is determined by the minimum spectral gap, the difference between the two lowest energy levels, of $\mathcal{H}(T)$. When the spectral gap is small, the Hamiltonian has to be evolved more slowly. To ensure a high success probability T needs to be lower bounded by $T = O\left(\frac{1}{g_{\text{min}}^2}\right)$ where g_{min} is the minimum spectral gap.

1.2 Problem encoding

Before solving any quantum or classical problem on a QA device it has to be encoded into a Hamiltonian that describes it. Hardware capabilities may limit which Hamiltonians we can encode, while the aim is to be able to encode arbitrary ones to reach universality by using more powerful hardware and algorithms. For example, a fermionic Hamiltonian can be encoded into a spin Hamiltonian with a Jordan–Wigner transformation. On the other hand, many classical combinatorial optimisation problems can be mapped into Quadratic Unconstrained Binary Optimisation (QUBO) [8], which only requires a resourceful but restricted type of Hamiltonian that has been already easily implemented, the classical Ising model. Solving these problems allows us to solve any other NP-complete problems as polynomial time mappings among them exist [8]. Let $\mathbb{B} = \{0, +1\}$ be the binary, $\mathbb{N}$ natural and $\mathbb{R}$ real sets. Given $\mathbb{B}$, $\mathbb{N}$ and $\mathbb{R}$, a general QUBO problem can be defined as a quadratic polynomial over binary variables with quadratic and linear terms:

$$f_Q(b) = \sum_i^n Q_{i,i} b_i + \sum_{i<j} Q_{i,j} b_i b_j \tag{2}$$

The QUBO binary variable $b \in \mathbb{B}^N (N \in \mathbb{N})$ forms the problem variable, the vector of $b_i \in \mathbb{B}$. Coefficients $Q_{ij} \in \mathbb{R}$ for $1 \leq j \leq i \leq n$ are QUBO parameters. Solving it means finding the vector b^* that minimises f_Q. Then the first step of the quantum formulation of an optimization problem is the mapping of its QUBO variables $b_i \in \{0, +1\}$ into the Ising binary variables $z_i \in \{-1, +1\}$. Applying the mapping $b_i \Rightarrow \frac{1-z_i}{2}$ yields its Ising equivalent (see appendix 6.1 for more details):

$$b^* = \underset{b \in \{0,1\}^n}{\arg\min} f_Q(b) \equiv \underset{z \in \{1,-1\}^n}{\arg\min} \sum_i h_i z_i + \sum_{i<j} J_{ij} z_i z_j \tag{3}$$

One typical example of QA Hamiltonian to minimise an Ising problem H_{final} would be:

$$\mathcal{H}(t) = \underbrace{-(1 - s(t)) \left(\sum_i \hat{\sigma}_x^i \right)}_{\text{Initial Hamiltonian}} + \underbrace{s(t) \left(\sum_i^n h_i \hat{\sigma}_z^i + \sum_{\langle ij \rangle} J_{i,j} \hat{\sigma}_z^i \hat{\sigma}_z^j \right)}_{\text{Final Hamiltonian}} \tag{4}$$

where $h_i, J_{ij} \in \mathbb{R}$ are the qubit biases and coupling strengths, respectively. $\hat{\sigma}_x^i = I \otimes I \otimes \ldots \otimes \sigma^x \otimes \ldots \otimes I \otimes I$ represents the Pauli matrix σ_x acting in the i^{th} position, and similarly for $\hat{\sigma}_z^i \hat{\sigma}_z^j$. The pairs of variables $\langle ij \rangle$ that take nonzero coefficients are called neighbours. $\mathcal{H}_{init}$ is usually chosen to comprise of transverse magnetic fields, $\mathcal{H}_{init} = -\sum_i \hat{\sigma}_x^i$, so that the ground state is an equal superposition of all states in the computational basis.

It is important to note that both classical and quantum problems can be regarded as a graph, $G = (N_G, E_G)$ (see appendix 6.2). For the classical Ising, each variable z_i represents a node $n_i \in N_G$ with weight h_i, and we draw an edge $e = \{z_i, z_j\} \in E_G$ if and only if its corresponding weight J_{ij} is nonzero. For arbitrary quantum models we proceed similarly but now keeping in mind that we have additional interactions and local terms σ_x, σ_y or σ_z that are also encoded as nodes and edges, respectively. The quantum chip can also be regarded as a graph, $D = (N_D, E_D)$, where each node $n_i \in N_D$ corresponds to a qubit, and each edge $ij \in E_D$ to the presence of a physical coupler between qubit i and j. Once both problem and chip graphs have been constructed they need to be mapped one to the other: this is called the graph embedding problem. However, mapping issues may appear due to discrepancies in the topology and type of connections required on both problem

and hardware graphs. These come from the fact that so far we can only fabricate devices with a limited density of connections due to engineering challenges. Moreover, for arbitrary quantum models we face additional difficulties related to the hardness of implementing σ_x and σ_y interactions in comparison with σ_z in certain hardware platforms.

1.3 Current approaches and limitations

There is still no favoured hardware implementation for QC, but common factors among them are the constraint on the number of connections each qubit can have and the lack of tunable interactions that go beyond Ising. However, for QA superconducting qubits [9] are preferred. In this platform, placing the qubits in planar chips makes it challenging to have high density of connections due to limited space and cross-talk. In other words, connectivity is far from all-to-all, being the identification of efficient graph embeddings on the available hardware topologies one of the main bottlenecks for QA.

The embedding of a graph involves finding the graph minor of G in D to act as a logical graph. This means that G can be embedded in D by deleting edges and nodes and contracting edges [10], so D has to be equal or bigger than G. However, deterministically finding it is an intractable problem. To deal with this, investigations have mainly focused on researching fast and high-quality heuristics, being the MinorMiner algorithm from Dwave one of the most popular ones [11]. This is a temporal solution for non-perfect graph embedding due to its non-scalability. For this mapping, each logical qubit is represented by a tree of ferromagnetically-coupled physical qubits called logical qubits or chain $T = \sum_i q_i$ for $i \in chain$ to keep them having equal binary values. Theoretically, by setting the inner chain coupling $J_{ij} = -\infty$, they ensure that q_i and q_j take the same value during the annealing, but current devices do not allow J_{ij} to take these values. Moreover, if chain strength is too large, the chains themselves will interfere and change the problem. Conversely, if it is too small, chains will have different values for each qubit and alter the problem. This creates the parameter setting problem for their chain embeddings [12], for which appropriate values for h_i and J_{ij} need to be set. Furthermore, for long chains it is challenging to find chain strengths powerful enough to balance other problem terms. For a summarised analysis of this algorithm see appendix 6.3.

Let us now explore the limitations of this approach. First, it is heuristic, so success is not guaranteed, neither is the non-existence of an embedding in case of failure. Second, each run can result in different embeddings, so the whole process is repeated and the best solution is selected, which is the one using fewer qubits and shorter chains. Third, it depends heavily on the fact that if G is smaller than D so there are numerous distinct G minors in D. Fourth, it is not scalable, as its time complexity is exponential, because it relies on a exponentially increasing number with N_D of shortest path distance calculations that need to be recomputed in each iteration due to the changing weights. Fifth, it is tailored to DWave's Chimera topology, which makes it quite restrictive, as this graph has a large treewidth and automorphism group, reducing the number of choices in their heuristic method [11].Moreover, the graph is sparse, meaning shortest paths can be computed in linear time, but for general graphs, the scaling of this shortest path search is worse. Sixth, we also found something odd regarding how it behaves with different hardware topologies. For the 1^{st} nearest neighbours of the grid graph, with degree 4, this approach will not find embeddings for anything larger than a complete 5 node graph K_5 even for huge grids. However, it does find embeddings for a less connected random graph of degree 3 (see appendix 6.3.4) and a 1^{st} and 2^{nd} nearest neighbours connected grid. Seventh, the

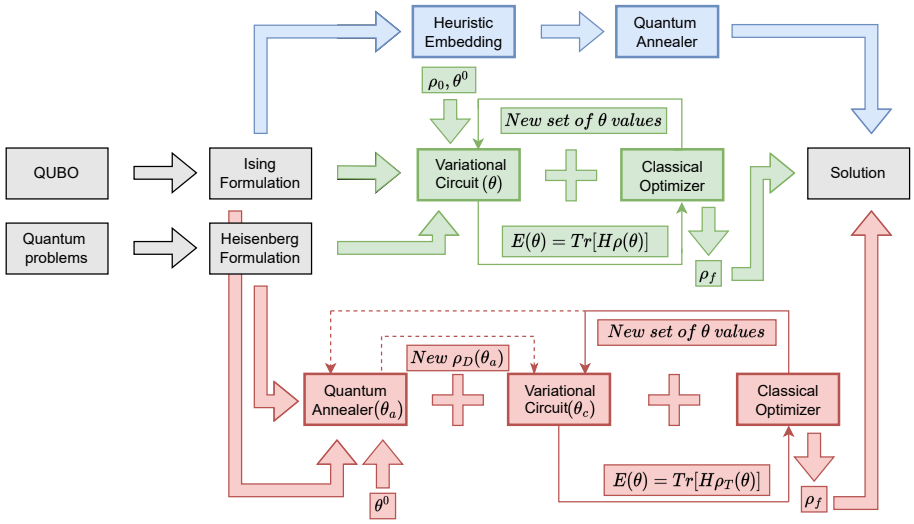

Figure 1: Workflow of the different NISQ approaches to solving QUBO (all) and quantum (green and red) problems. The red dashed line belongs to DaTA and is what differences it from DaFA.

heuristic is restricted to embeddings into Ising models, so it can not work with arbitrary quantum problems. Finally, for the lab implementation the following issues, derived from this embedding strategy, are detrimental to performance [13]. First, the required precision and control over the setting up of h_i and J_{ij} makes the problem subject to control errors. Second, limitations on the tunability of h_i and J_{ij} relative to thermal scales K_bT bound the range of parameters that can be used in the search for an embedding. That poses a problem for systems with very small gaps, where very low temperatures, currently unreachable, would be required in order to obtain an acceptable embedding solution.

1.4 Novel approach

In the pursuit of universality in the NISQ era, we choose a hybrid approach that allows us to target any desired Hamiltonian. For this we take the most of the two main models of QC and propose a hybrid Digitally-assisted Tunable/Fixed Annealing (DaTA/DaFA) or more generally, Digitally-assisted Annealing (DaA) algorithm (red flow of Fig. 1). We use QA as a first part of our approach as it is advantageous over gate QC for being less prone to control errors. We then explore the use of variational quantum circuits inspired on VQE [14](green flow of Fig. 1) to fix the connectivity issues and the lack of certain interaction types on our device. The novelty of those approaches rely on the fact that the state generated by the quantum annealer is used as the initial state of the variational circuit, the role of which is to approach a target solution the annealer could not reach by itself due to its hardware limitations. We design this approach with hopes of improving the reach, efficiency and accuracy of heuristic-based embeddings (blue flow of Fig. 1).

2 Variational Quantum Eigensolver, VQE

VQE is a variational hybrid quantum-classical algorithm [15, 16] where a parameterised quantum circuit or ansatz $(U_L(\theta_L)\dots U_1(\theta_1))$ is applied to an initial state ρ_0 to prepare a complex density matrix $\rho(\theta) = U_L(\theta_L)\dots U_1(\theta_1)\,\rho_0 U_1^\dagger(\theta_1)\dots U_L^\dagger(\theta_L)$ by sequentially minimising a classical cost function that is constructed out of measured quantum observables. Its goal is typically to compute the ground state of a Hamiltonian H and θ is the

collection of parameters that describe the parameterised gate operations of the circuit, $U(\theta)$. VQE is commonly used for Quantum Chemistry applications like finding the ground state of molecular systems. Its characterisation, gate placement and operation order, is purely classical, subject to numerical optimisation with classical assistance. VQE uses the variational principle which states that the average energy of the ansatz is strictly larger or equal than the energy of the sought ground state, $\lambda_{\min} \leq E(\theta)$ to build its cost function $C(\theta) = E(\theta) = \mathrm{Tr}(\rho(\theta)H)$ as the expected energy, which is averaged over all states. Solving for $\theta_{\text{opt}} = \text{argmin}_\theta\, E(\theta)$ gets a good ground state approximation.

However, VQE faces the exponential size of the Hilbert space that makes any approach that searches paths characterising the quantum circuit in the parameter space handle tiny gradients. This may cause classical optimisers to get stuck on local minima or lost, especially with large problems and more complex energy landscapes. Another limitation is circuit depth coming from problems that may require so many layers that they go beyond the coherence capabilities of the devices. On the other hand, QA is guaranteed to find the ground state of a Hamiltonian, at the expense of a very slow evolution. Both methods' pros and cons have been reviewed in [17], and here we aim at combining the advantages of both to solve the graph discrepancies between the problem and hardware, as well as for engineering missing interactions not present in the hardware that the problem may require.

3 Digitally-assisted Annealing, DaA

We propose DaA as an extension of VQE that uses adiabatic or diabatic evolution to improve efficiency and get rid of the hardware limitations in terms of connectivity and interaction types of using typical QA alone. It is an analogue-digital algorithm because we tune the chip that uses QA (analogue model) to generate initial states for the VQE (digital model). The main difference between our algorithm and traditional QA is that the lack of certain type of interactions and density of connections is handled by the digital part while the analogue bit allows us to prepare a favourable initial state ρ_D. That way, we also hope to make the required circuits smaller while reducing the accumulated errors that a high amount of gates introduce in NISQ devices. In this method, we first create the initial Hamiltonian H_{init} of the QA algorithm which is implementable in an available Ising-based annealer that is to be driven to an H_{final}, defined in the hardware graph D, whose parameters h_i and J_{ij} are tunable for DaTA and fixed for DaFA. That means, DaFA is the same as DaTA but without optimisation happening in the annealer. So at every step, the parameters θ divided on those from the circuit, θ_c, and those from chip Hamiltonian , θ_a, are optimised classically for DaTA while for DaFA that only happens for θ_c. At first, the parameters are just picked randomly from a uniform distribution over a given interval. Our aim is to modify H_{final}, which we will just call H form now on, such that after applying a complex enough $U(\theta)$ its eigenstates $|\eta_k\rangle$ with eigenvalue η_k are mapped to the arbitrary problem H_{target}, defined as the graph G, with eigenstates $|\lambda_k\rangle$ and eigenvalues λ_k. It is important to note that for degenerate cases the optimal $U(\theta)$ will not be unique, as eigenstates need only be mapped in the degenerate target subspace. The general outline of the method is as follows:

1. Use the annealer to prepare a thermal state $\rho_D(\theta_a)$.
2. Apply the parameterised quantum circuit $U(\theta_c)\rho_D(\theta_a)U^\dagger(\theta_c)$.
3. Measure our cost function: $C(\theta) = \mathrm{Tr}[H_{target}U(\theta_c)\rho_D(\theta_a)U^\dagger(\theta_c)]$

4. Optimise $C(\theta)$ by tuning θ in DaTA or just θ_c on DaFA with the classical optimiser of choice (see Appendix 6.5).

We assume ρ_D to be well described by a Gibbs state $\rho_H = \frac{1}{Z}\exp[-\beta H]$ with $\beta = \frac{1}{K_B T}$. The (effective) inverse temperature β gives us a measure of how noisy the process is. For a quantum mechanical discrete canonical ensemble, the partition function Z is defined as the trace of the Boltzmann factor: $Z = \mathrm{tr}(e^{-\beta H})$. Let us note that DaA works for any arbitrary problem and encoding unlike DWave's approach, i.e. our H_{target} can take the form of any arbitrary Hamiltonian. In order to evaluate the performance of our method, we conduct the experiments on a classical simulator of an ideal quantum computer in the limit of infinite measurements, eliminating quantum uncertainty. These results can be complemented in the future by simulating a finite amount of measurements emulating QC indeterminacy. The code used Networkx [18] for the representation of graphs and Qibo [19] for quantum circuits and optimization. As we only analysed small circuits, less than 10 qubits, the overhead from casting arrays to GPU was larger than just executing on CPU.

3.1 Hardware and target problem details

Current quantum devices have limited capabilities, so identifying hardware-efficient approaches and algorithms that make the most out of them is crucial. We take this into consideration and in our simulations we do not allow the presence of any two-qubit gates that cannot be implemented via the native chip connectivity, something that would require costly additional SWAP operations. To leave more room for the optimization we generally assume the hardware chip to have more qubits than required by the problem to be encoded. The additional qubits are ancillary qubits identified with the subscript A and they will introduce degeneracy to the system after the embedding. We then use the subscript T to denote the target qubits upon which G is acting. We note that as long as we are consistent with the qubits' labelling in the cost function and partial trace, we can choose any qubits as target ones. Tr_A and Tr_T stand for the partial traces over these respective degrees of freedom. After tracing out the ancillary system, we end up with a reduced density matrix we called effective Hamiltonian H_{eff}, defined by:

$$\rho_T(\theta) = \mathrm{Tr}_A\left[U(\theta)\rho_D U^\dagger(\theta)\right] \sim \exp[-H_{eff}(\theta)] \tag{5}$$

We can conceive H_{eff} as the Hamiltonian acting on sub-system T after tracing the ancillas A. The cost function $C(\theta)$ takes into account this discrepancy in dimensions as follows:

$$C(\theta) = \mathrm{Tr}\left[U(\theta)\rho_D U^\dagger(\theta)(H_{target}\otimes \mathbb{1}_A)\right] = \mathrm{Tr}\left[\mathrm{Tr}_A[U(\theta)\rho_D U^\dagger(\theta)]H_{target}\right] \tag{6}$$

We will benchmark this approach asking two different questions. First, can our method allow us to simulate more complex connectivities than the device's one? And second, can our method simulate problems encoded in Hamiltonians that go beyond the Ising model? As Ising interactions are relatively easy to engineer in superconducting flux qubit devices, one of the most common current implementations of quantum annealers. For this we assume our hardware to be of this type and be described by the following Hamiltonian:

$$H_{Hardware} = \sum_{i\in V} h_i z_i + \sum_{i,j\in E} J_{ij} z_i z_j \qquad\qquad z_i = (1-\sigma_i^z)/2 \tag{7}$$

We consider two problems with different associated Hamiltonians: a classical Weighted Maximum Independent Set (WMIS) problem (see appendix 6.6) and the simulation of an

arbitrary quantum XX-YY-Heisenberg model:

$$H_{classical} = \sum_{i \in V} h_i z_i + \sum_{i,j \in E} J_{ij} z_i z_j \tag{8}$$

$$H_{quantum} = \sum_{i \in V} h_i z_i + \sum_{i,j \in E} J_{ij} x_i x_j + J_{ij} y_i y_j \tag{9}$$

3.2 Circuit ansatzes

We want our circuits to be highly expressive using the minimal number of gates to reduce the impact of errors [20]. To attain this, it is important that the circuit is chip-motivated and for that we introduce what is known as hardware-efficient ansatzes. These ansatzes include two-qubit gates that are part of the native gate set, i.e. two-qubit gates are applied only between qubits pairs that have an existing physical link. This allows to maintain the circuit depth, as no additional SWAP or Bridge gates to logically connect non-physically linked qubits are required. This is not the only way to construct ansatzes given that those are often problem-dependent, with different ways to organise the gates and their parametrisation [20][18]. For our analysis the hardware-efficient ansatz is enough to benchmark DaTA vs DaFA to give a solution to the lack of connectivity and interactions available.

After these considerations, we build the circuit from single-qubit parameterised gates and a fixed two-qubit gate blocks. One sequence of blocks is called a layer, and usually, several of them are used. For the single-qubit gates we choose parameterised general single-qubit rotations U_3 and for the two-qubit gates, Controlled-NOT gates (CNOT). For each block, see Fig. 2, a), we place 4 U_3 gates, with a total of 12 different parameters to optimise per block. For our ansatz, we observe that by combining the consecutive single-qubit rotations into just one the performance improves from reducing its parameter count without any downsides. We called this the reduced blocks ansatz that is shown in Fig. 2, b). The effect of alternating control and target qubits among layers for small-scale problems was also studied, but we did not appreciate any relevant changes.

$$U_3(\theta, \phi, \lambda) = \begin{pmatrix} e^{-i(\phi+\lambda)/2} \cos\left(\frac{\theta}{2}\right) & -e^{-i(\phi-\lambda)/2} \sin\left(\frac{\theta}{2}\right) \\ e^{i(\phi-\lambda)/2} \sin\left(\frac{\theta}{2}\right) & e^{i(\phi+\lambda)/2} \cos\left(\frac{\theta}{2}\right) \end{pmatrix} \qquad \text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

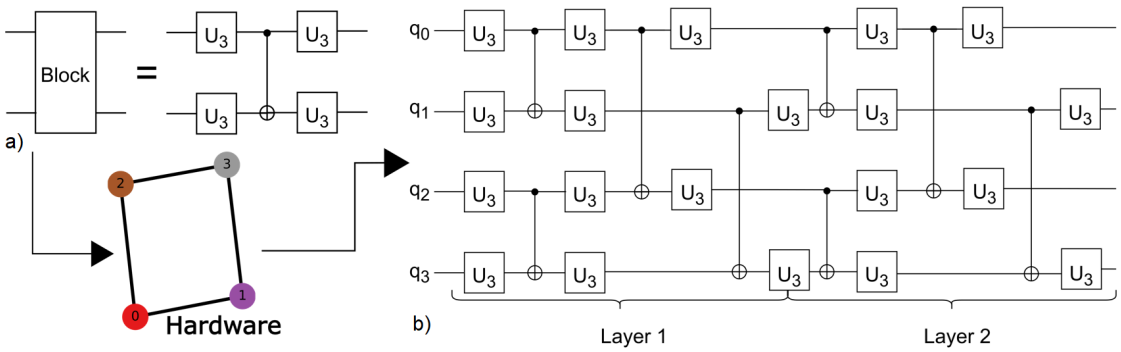

Figure 2: a) Building block for the variational circuit. b) Variational layers for a 2x2 1^{st} nearest neighbours grid Fig. 3 d) for the reduced block ansatz with an average of 33 parameters per layer.

3.3 Efficiency indicators

To evaluate performance we study two different efficiency indicators. First, we evaluate the state probability, P_i, as the probability of measuring a state belonging to the ground ($i = 0$) or one of the excited subspaces ($i > 0$). Then, the state fidelity, F_i, which measures how close our states, ground ($i = 0$) or excited ($i > 0$), are to the target ones:

$$P_i = \mathrm{Tr}\left[\rho_T \epsilon_i\right] \qquad\qquad F_i = \langle \tau_i | \epsilon_i | \tau_i \rangle \tag{10}$$

where $\epsilon_i = \sum_k |\lambda_i^k\rangle\langle\lambda_i^k|$ represents the degenerate target subspace i, and $|\tau_i\rangle$ are the eigenvectors extracted from ρ_T with eigenvalues τ_i arranged in ascending order $\tau_i \leq \tau_{i+1}$. We assess both indicators for the ground state by plotting P_0 and F_0 alongside the normalised cost function $C_N(\theta) = \frac{C(\theta)+|\lambda_0|}{|\lambda_0|+|\lambda_{max}|}$ versus the amount of circuit layers of our ansatz. We also represent the final thermal state $\rho_F = \sum_i \frac{\sum_l^L P_i * |\tau_{il}\rangle\langle\tau_{il}|}{L}$, where L is the degeneracy of each target subspace, in a histogram alongside with the target solution $|\lambda_0\rangle$ or solutions $\sum_k^K |\lambda_0^k\rangle/K$ (here assumed to be known) where K is the degeneracy. The algorithm's success is also quantified by the computational basis bitstring probability distribution and how it overlaps with the target solution. For Ising problems, to retrieve the solution we just need to find the most probable state through measurement statistics in the Z basis, as the solution is a single bitstring. However, for quantum Hamiltonians even it the bitstring probabilities match with the target ones, the state may not be correct due to the lack of phase information. Nevertheless, this match is a good sign, but for the full state information we need to look into measurements in a different basis. In the next section, we will use these metrics to compare the performance of DaFA and DaTA algorithms for different Hamiltonians, degenerate and non-degenerate target ground states and different sizes of both the problem and the chip.

4 Results and discussion

Let us now move to the results of the variational optimisation and their analysis. All the results here have been averaged over 5 realisations and their uncertainties are shown as error bars, sometimes too small to be noticed. For each of these 5 realisations we initialise both the circuit and the chip with a set of random values chosen uniformly from $[-2\pi, 2\pi]$. We label the number of layers of the circuit, i, as d_i and the total number of required parameters, j, as θ_j. Then we refer to K_i as a complete graph (i.e. all-to-all connected), R_i as a randomly generated graph with $2/3$ probability of edge creation, G_i as a grid graph and C_i as a chain graph, being i the number of nodes. Both the chip and problem graphs will be showed next to the the F_0 and P_0 for clarity. The weights of the target problem are taken to be integers selected randomly from the following uniform distributions: $h_i \in \{-2, -1, 0\}$ for the nodes, $J_{ij} = 2$ for the K_i edges and $J_{ij} \in \{1, 2\}$ for R_i edges. The initial chip for DaTA and fixed chip for DaFA is parameterised like a K_i graph. Finally, we bound the parameters of the chip Hamiltonian (considered to be an Ising Hamiltonian), $h_i, J_{ij}, \in \{-3, 3\}$. Finally, the plotted lines were displaced slightly on their corresponding X and Y axis to avoid overlapping with the others.

4.1 Temperature dependence on the optimisation

We start by analysing the effect of temperature by changing β, which is the inverse temperature, for the simple case displayed in Fig. 3. In this analysis we review three different scenarios: $\beta = 5$, $\beta = 1$ and $\beta = 0.25$. For $\beta = 5$ the ground state probability approaches

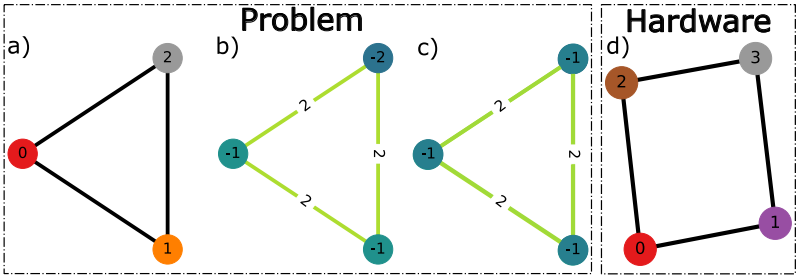

Figure 3: a) K_3 target problem with b) non-degenerate and c) degenerate ground-state solutions. d) Hardware layout, G_4

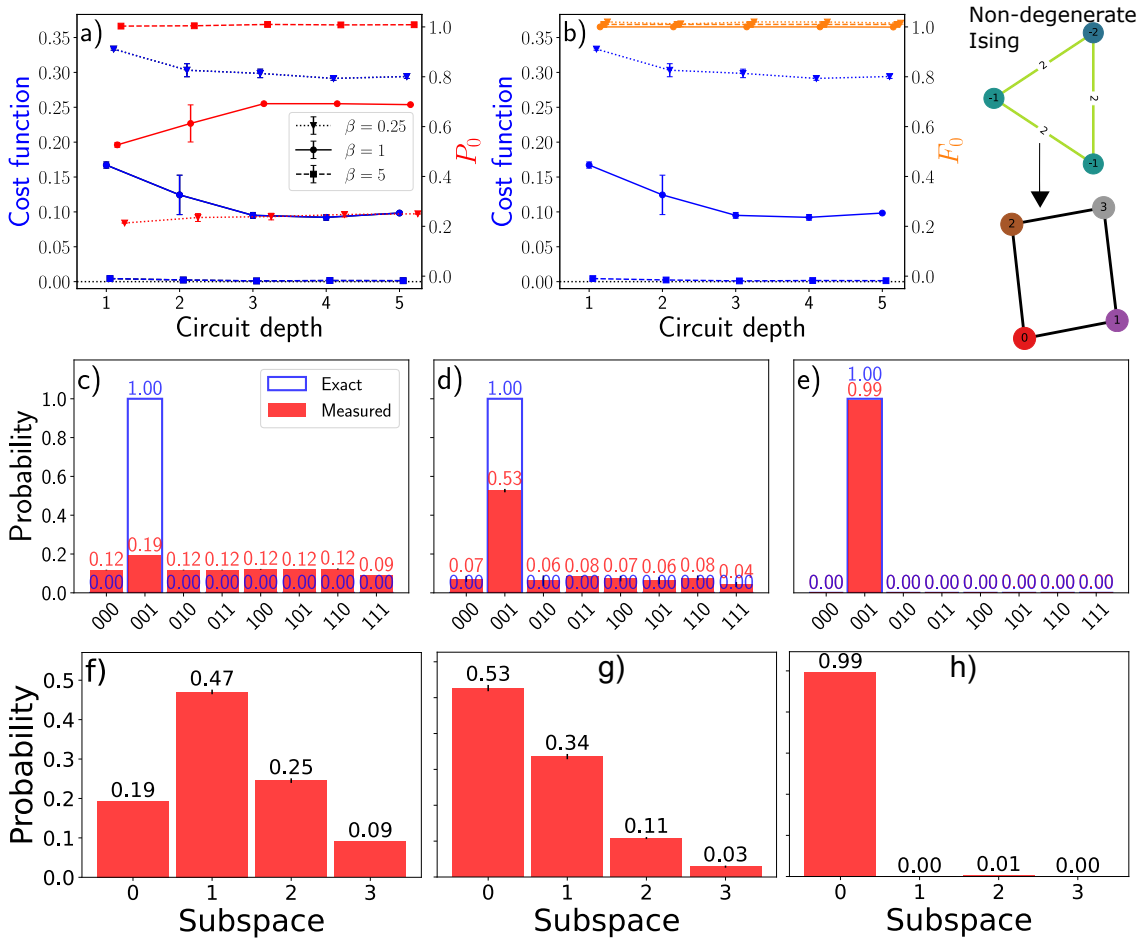

Figure 4: Temperature analysis for a non-degenerate solution K_3 Ising target Hamiltonian to be embedded into G_4 hardware graph using DaFA for d_1 and θ_{36}. a) P_0 b) F_0. All bitstring overlaps and P_i for c),f) $\beta = 0.25$; d),g) $\beta = 1$ and e),h) $\beta = 5$, respectively

the unity, for $\beta = 1$ the ground state predominates over all the other excited states and for $\beta = 0.25$ some excited states are more probable than the ground state. The tests are first conducted using DaFA for non-degenerate K_3 Ising target problems to be embedded into G_4. What we observe in Fig. 4, b) is that for any of the temperature values considered we manage to learn the ground state with high fidelity, in spite of thermal noise. In Fig. 4 a), f) and g) we observe clearly how the result comes from sampling the noisy state and not just the ground state of our device. We now compare the performance of DaTA and DaFA for a non-degenerate K_3 Ising and Heisenberg target Hamiltonian (a) and b) in

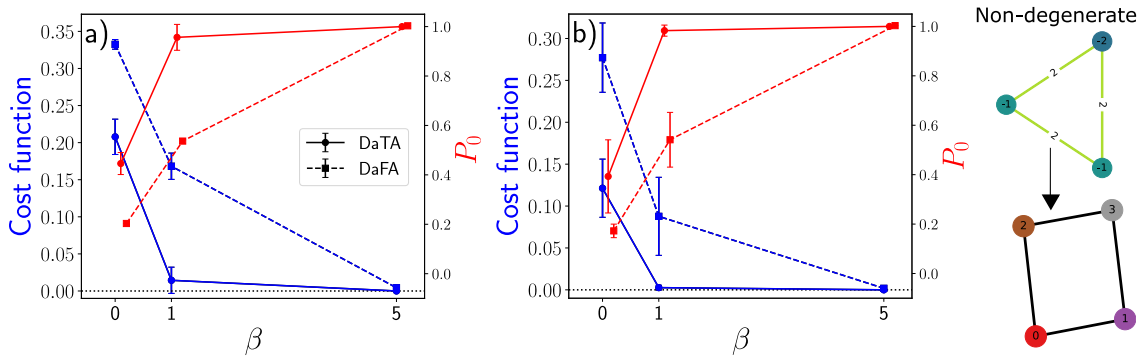

Figure 5: DaTA vs DaFA for non-degenerate K_3 into G_4 with variable temperature. a) P_0 for Ising with DaTA (d_1, θ_{44}) vs DaFA (d_1,θ_{36}), b) P_0 for Heisenberg with DaTA (d_2, θ_{68}) vs DaFA (d_2,θ_{60}), deeper circuit because this is a harder case that requires it to start behaving optimally.

Fig. 5, respectively) that has been embedded into a G_4 hardware graph for different β and observe similar behaviour for both Hamiltonians provided we have enough circuit layers: as we increase inverse temperature the ground-state probability increases. Importantly, we observe that DaTA learns to reduce the thermal noise by maxing out the parameter ranges. For the rest of the results, we choose $\beta = 1$ as it yields good results while still considering the presence of moderate thermal noise that makes our investigations more realistic.

4.2 Simple K_3 target problem case

Now, we analyse the performance of DaA for both Hamiltonians and different degeneracy.

4.2.1 Non-degenerate Ising and Heisenberg target Hamiltonians

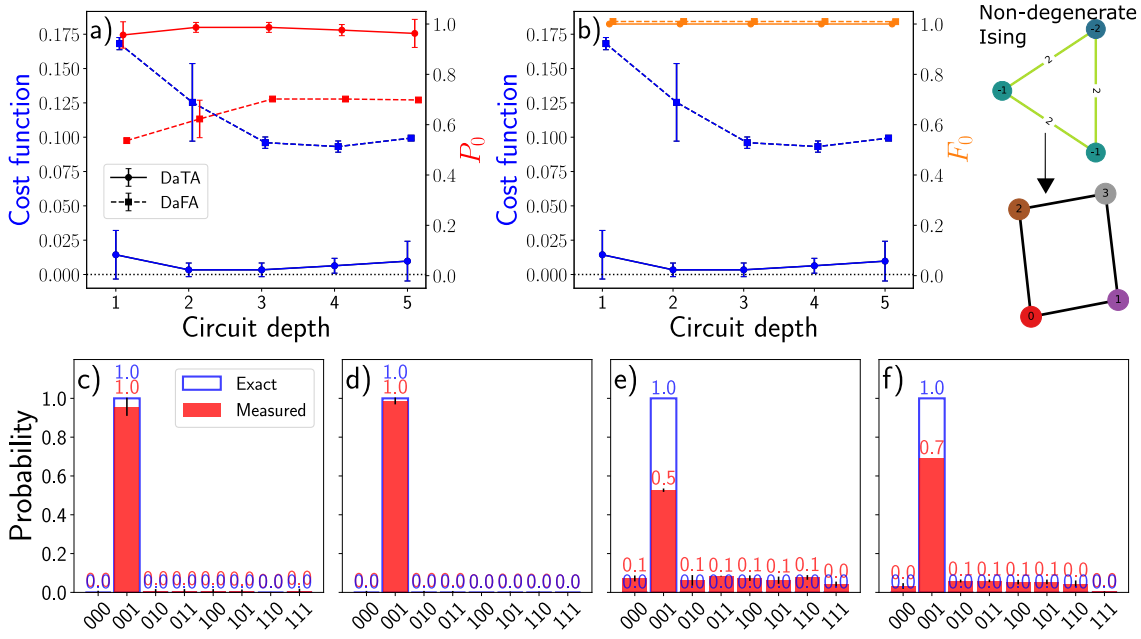

Figure 6: DaTA vs DaFA for non-degenerate K_3 Ising into G_4. a) P_0 b) F_0. All bitstrings overlaps for c) DaTA with (d_1, θ_{44}), d) DaTA with (d_3, θ_{92}), e) DaFA with (d_1, θ_{36}), f) DaFA with (d_3, θ_{84}).

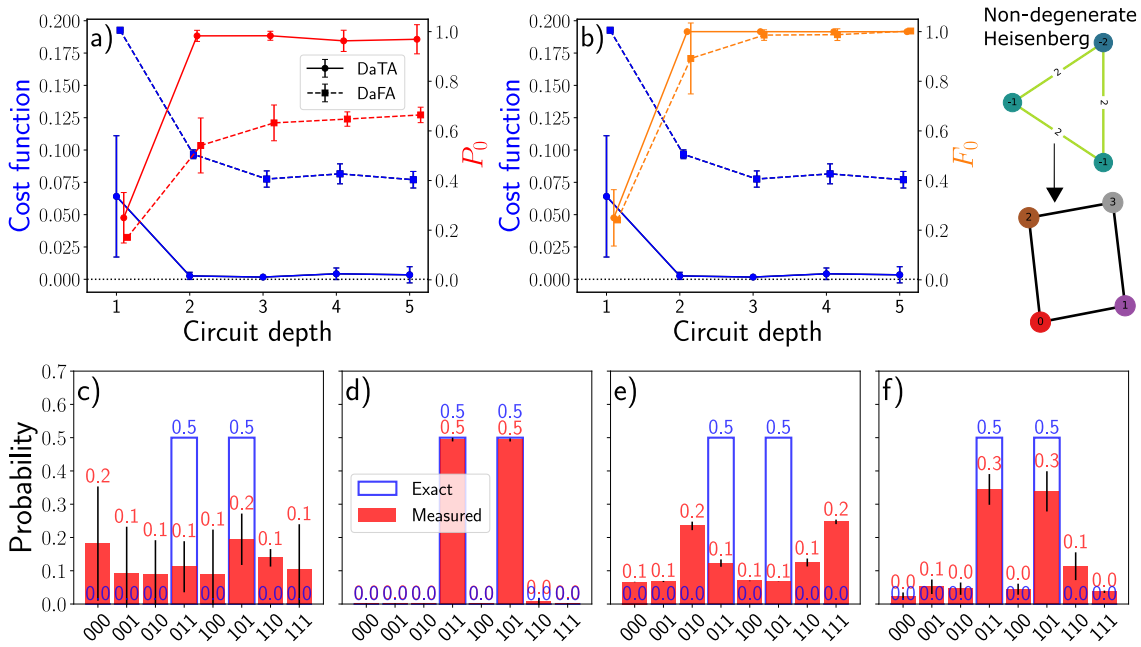

Figure 7: DaTA vs DaFA for non-degenerate K_3 Heisenberg into G_4. a) P_0 b) F_0. All bitstrings overlaps for c) DaTA with (d_1, θ_{48}), d) DaTA with (d_3, θ_{92}), e) DaFA with (d_1, θ_{36}), f) DaFA with (d_3, θ_{84}).

We start comparing the performance of our approaches for target problems with non-degenerate solutions. For the Ising case in Fig. 6, DaTA gives slightly better overlaps than DaFA for any circuit depth. However, DaFA is generally faster in terms of number of optimisation steps required to converge. If we move to the Heisenberg case in Fig. 7, we observe a big performance increase for d_2 with both approaches. We believe this is because generally the circuit needs to be more complex to go from the chip ZZ interactions to XX and YY ones. DaTA thermal noise removal is key when working with quantum Hamiltonians because it allows DaTA to get an exact result with high probability of measuring it while DaFA only manages to get a noisy result, as shown in Fig. 7.

4.2.2 Effect of degeneracy in the solution

We conduct an analogous analysis as the previous one but with degenerate solutions. In Fig. 8, the problem ground state is triple degenerate, each solution corresponding to equally likely single bistrings whose fidelities are represented with different orange tonalities in inset b). We highlight that for circuit layers larger than d_3 DaFA reaches enough expressability to map all 3 solutions, while this happens at d_4 for DaTA. Here we observe that once the performance indicators have saturated there is not point in increasing the circuit depth further as the optimisation becomes faultier as shown in Fig. 8 e),f) and in the slight decrease of F_0. For the Heisenberg case, the problem ground-state is double degenerate with contributions from several bitstrings as they are superposition states. Now, both algorithms require d_4 to return useful overlaps as we see in Fig. 9, b). However, as in the last Heisenberg case, DaTA returns the solution state with high probability while DaFA carries the thermal noise. To conclude, degenerate cases pose a greater challenge as the system needs to learn more states, but it is ultimately able to do so.

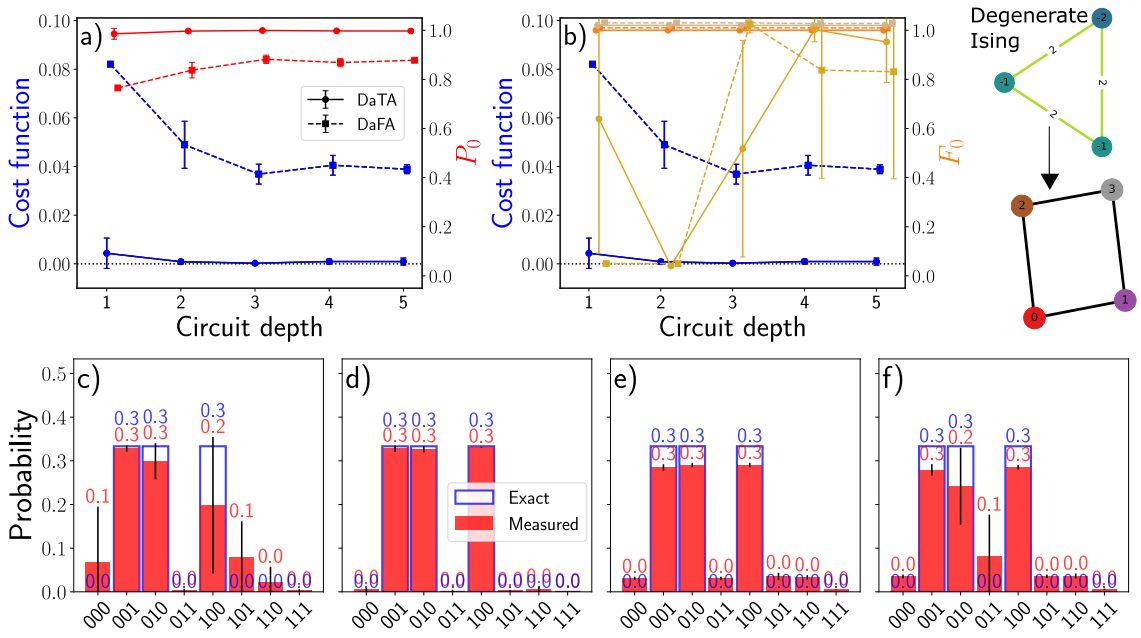

Figure 8: DaTA vs DaFA for degenerate K_3 Ising into G_4. a) P_0 b) F_0 for the 3 degenerate states (different orange shades). All bitstrings overlaps for c) DaTA (d_3, θ_{92}), d) DaTA with (d_4, θ_{116}), e) DaFA with (d_3, θ_{84}), f) DaFA with (d_4, θ_{108}).

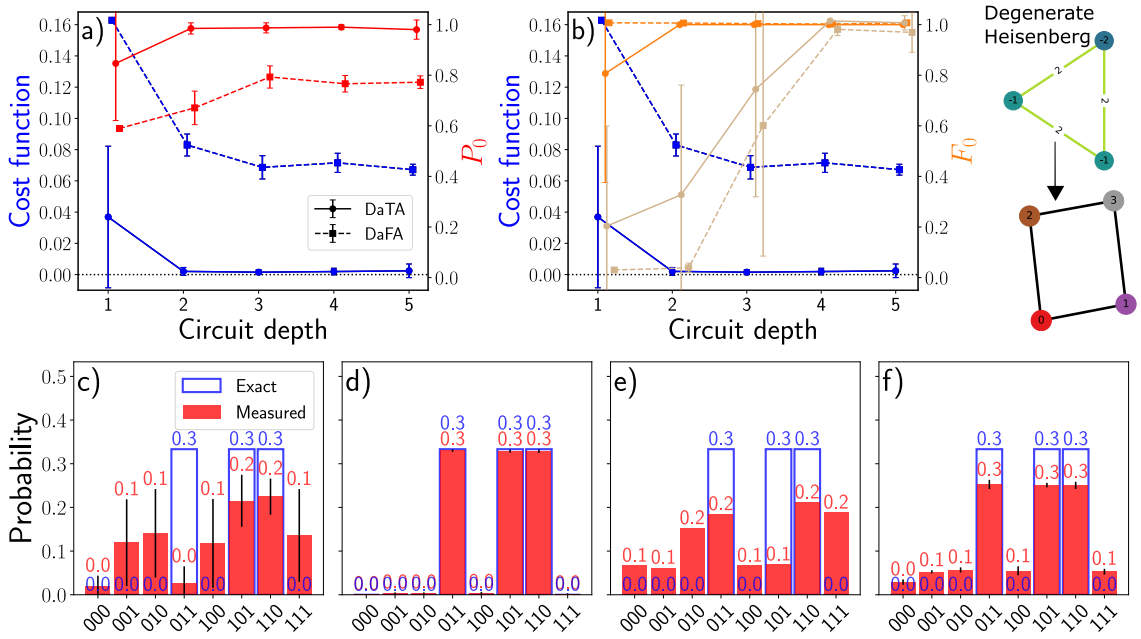

Figure 9: DaTA vs DaFA for degenerate Heisenberg K_3 into G_4. a) P_0 b) F_0 for the 2 degenerate states (different orange shades). All bitstrings overlaps for c) DaTA with (d_1, θ_{44}), d) DaTA with (d_4, θ_{116}), e) DaFA with (d_1, θ_{36}), f) DaFA with (d_4, θ_{108}).

4.3 Scaling of the problem and different topologies

We present the chips and problems graphs considered for the scaling analysis in Fig. 10. Note that as we scale the problem the edge count increases at a faster rate than the node count, and this is where the hardness comes from. It is useful to have in mind that K_4 is

composed of 6 edges, R_5 of 8, R_6 of 11 and R_7 of 14. On the other hand, G_5 only has 5 edges; G_6, 7; G_7, 8 and C_7, 6.

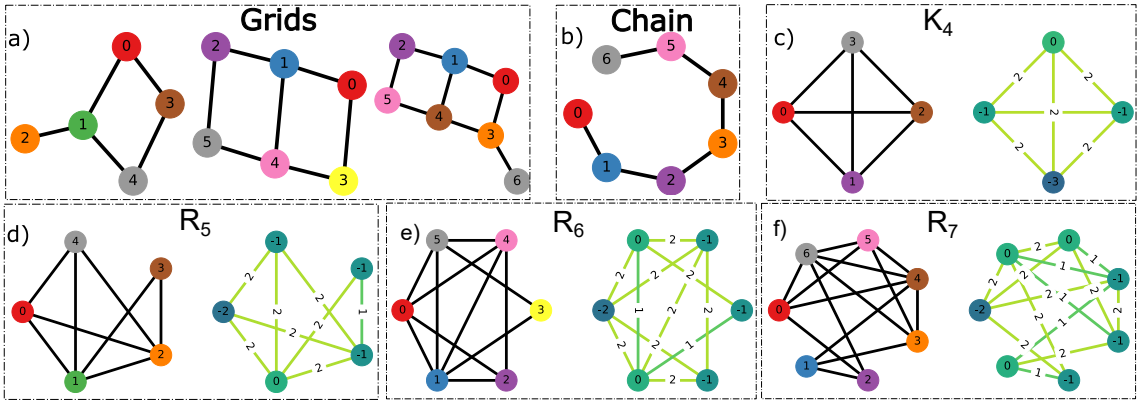

Figure 10: Labelled chip topologies for a) G_5, G_6, and G_7 b) C_7. Labelled (left) and weighted (right) non-degenerate solution problem graphs for c) K_4 d) R_5, e) R_6, f) R_7. Note, f) has different weights for non-degenerate Ising case.

4.3.1 Scaling up the Chip

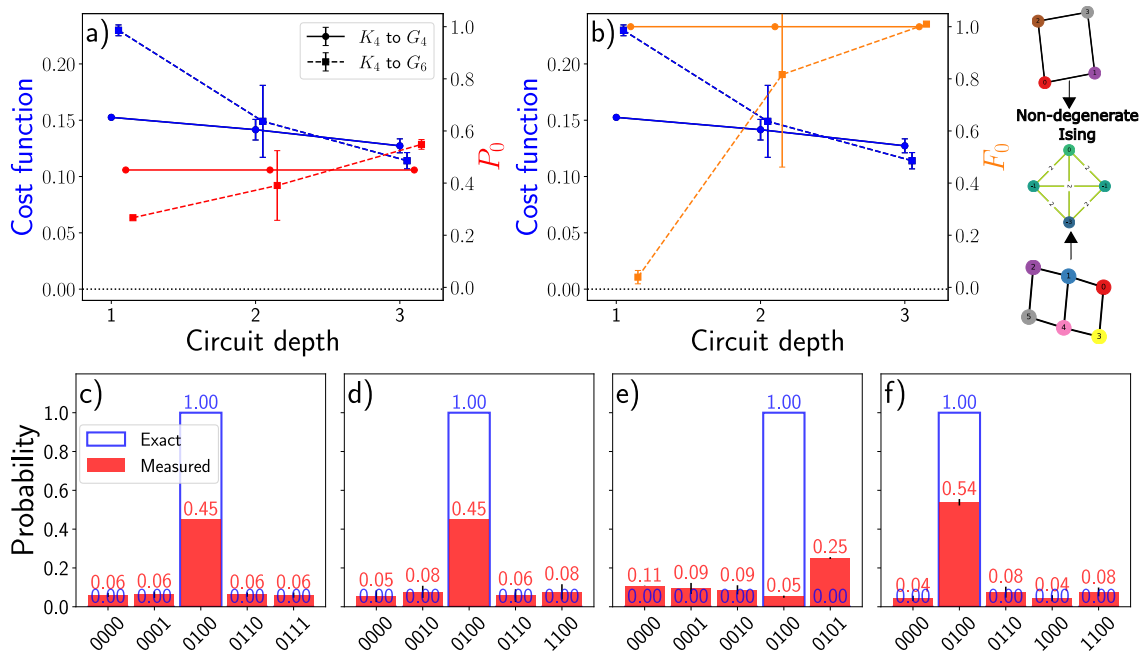

Figure 11: G_4 vs G_6 Ptr[0-1] for non-degenerate K_4 Ising using DaFA. a) P_0 b) F_0. Most probable 5 bitstrings overlaps for c)G_4 with (d_1, θ_{36}), d) G_4 with (d_3, θ_{84}), e) G_6 with (d_1, θ_{63}) and f) G_6 with (d_3, θ_{147}).

First, we study the chip size effect. For this purpose, we fix our problem to be K_4 into G_i chips. In Fig. 11 we compare the embedding of the problem into G_4 with G_6 with DaFA. F_0 for the G_4 quickly reaches unity, while G_6 only gets good values at d_3. We find that the performance of P_0 is moderate for both graphs, but the probability of the larger graph tends to increase with circuit depth. Nevertheless, DaTA performs much better for these cases: in Fig. 12 the smaller grid performs so well that it leaves no room for improvement

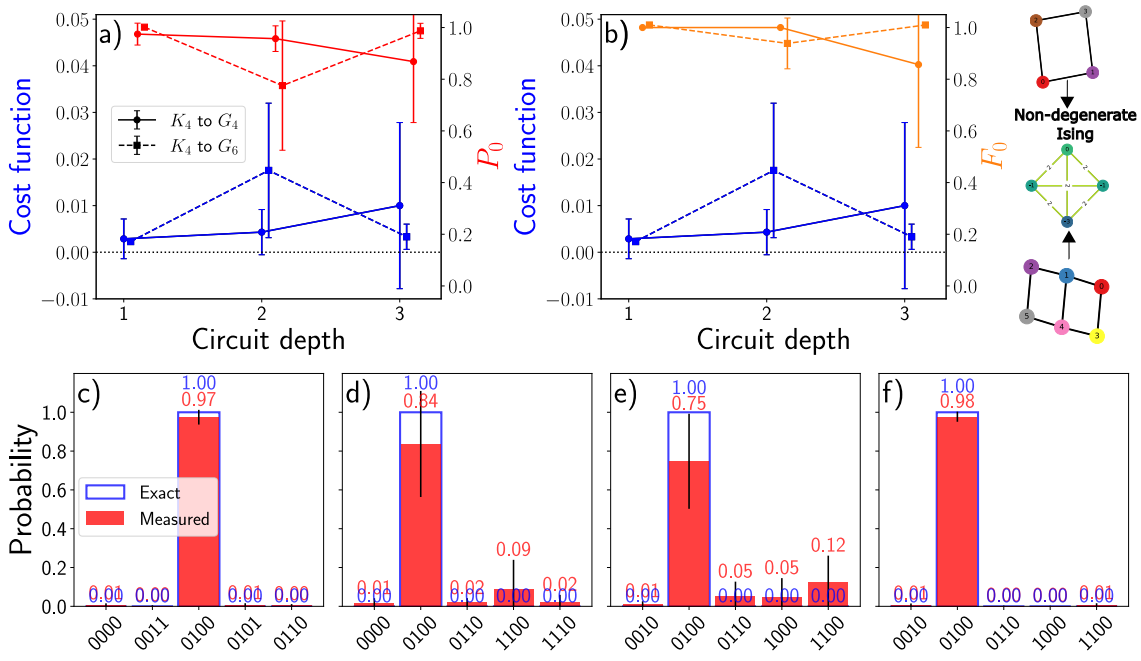

Figure 12: G_4 vs G_6 Ptr[0-1] for non-degenerate K_4 Ising using DaTA. a) P_0 b) F_0. Most probable 5 bitstrings overlaps for c) G_4 with (d_1, θ_{44}), d) G_4 with (d_3, θ_{92}), e) G_6 with (d_2, θ_{118}) and f) G_6 with (d_3, θ_{160}).

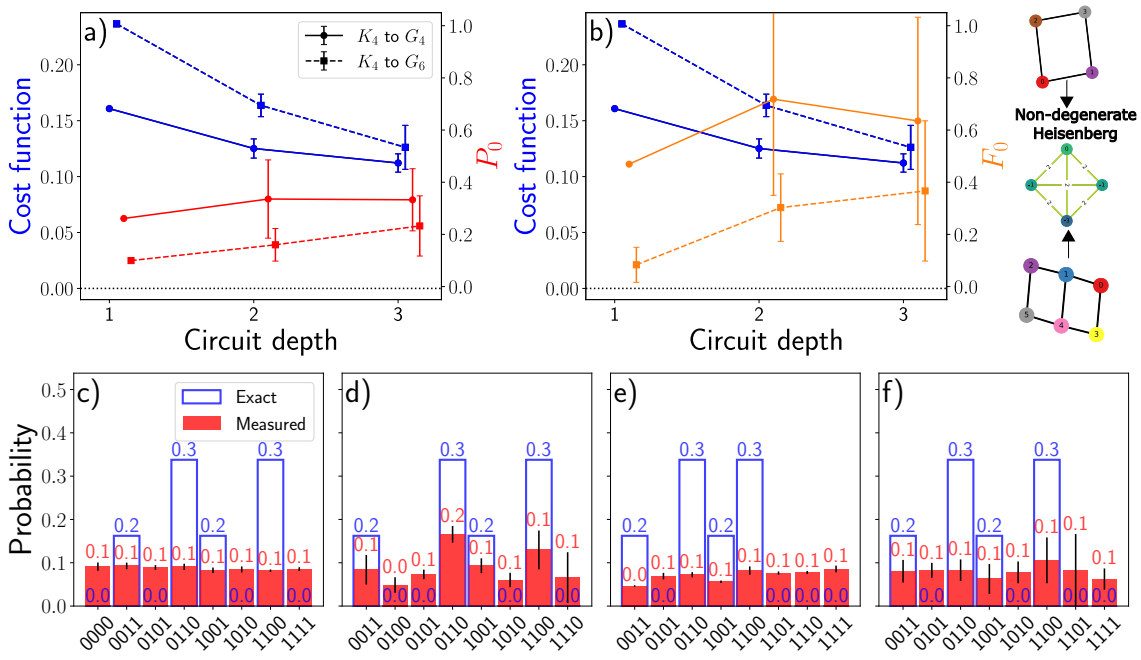

Figure 13: G_4 vs G_6 Ptr[0-1] for non-degenerate K_4 Heisenberg using DaFA. a) P_0 b) F_0. Most probable 8 bitstrings overlaps for c)G_4 with (d_1, θ_{36}), d) G_4 with (d_3, θ_{84}), e) G_6 with (d_1, θ_{63}) and f) G_6 with (d_3, θ_{147}).

for the larger one and they end up performing similarly, aside from statistical variations. Moving on to the harder Heisenberg case, we show in Fig. 13 that DaFA is not capable of finding the solution for the limited circuit depths considered, neither can we see the bigger system perform better. Meanwhile, with DaTA (see Fig. 14) we almost find an exact

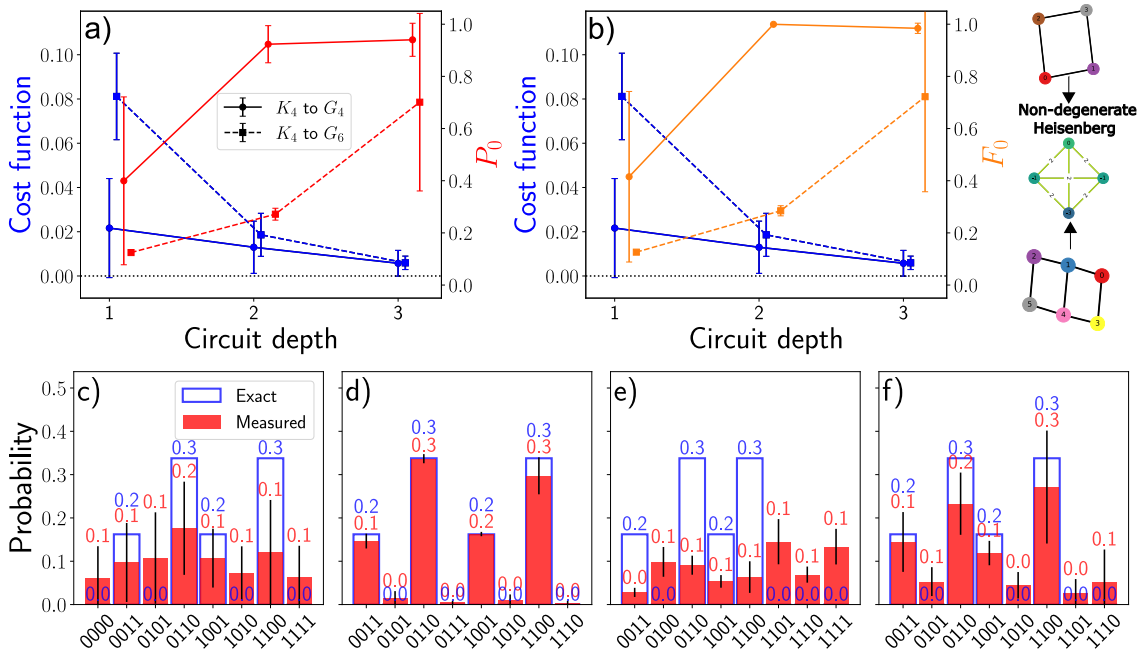

Figure 14: G_4 vs G_6 Ptr[0-1] for K_4 non-degenerate Heisenberg using DaTA. a) P_0 b) F_0. Most probable 8 bitstrings overlaps for c) G_4 with (d_1, θ_{44}), d) G_4 with (d_3, θ_{92}), e) G_6 with (d_1, θ_{76}) and f) G_6 with (d_3, θ_{160}).

solution with G_4 while for G_6, if the tendency is kept, an extra layer would be needed to match the same performance. We also note how similar looking P_0 and F_0 are due to DaTA's temperature-decreasing effect, which will also be observed in many of the following DaTA graphs. To sum up, DaTA keeps outperforming DaFA in this scenario while the increase of ancillary qubits in the chip seems to affect negatively the performance of the optimisation but further investigations are required to draw definitive conclusions.

4.3.2 Scaling up the Problem

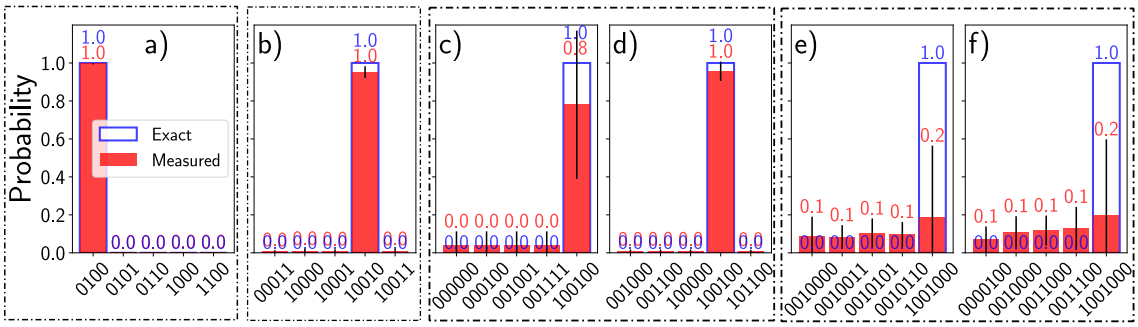

Figure 15: Non-degenerate Ising problems mapped into G_7(Ptr[1,5,6], Ptr[0,1], Ptr[1]) with DaTA. Most probable 5 bitstrings overlaps with (d_1, θ_{87}) on a) K_4, b) R_5, c) R_6, e) R_7; (d_3, θ_{183}) on d) R_6, f) R_7.

We now move on to the analyse the effect of problem size. In Fig. 15 we consider the Ising case for DaTA, G_7 as the hardware graph and the problem graphs are shown in Fig. 10 c), d), e) and f). For the small problems (K_4, R_5) we already reach good solutions with d_1, while for R_6 the same probabilities require deeper circuits and for R_7 we are not able to recover the solution due to the stark connectivity difference. We perform a similar analysis

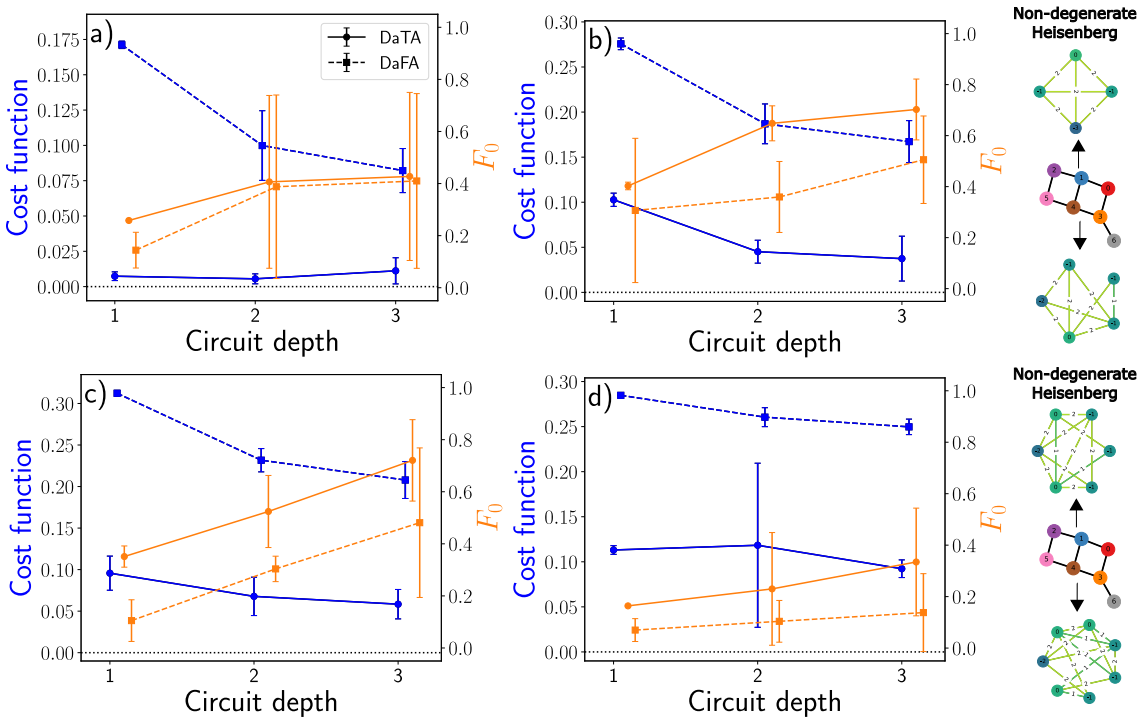

Figure 16: Non-degenerate Heisenberg problems into G_7(Ptr[0,5,6], Ptr[0,1], Ptr[1]) with DaFA vs DaTA. F_0 for a) K_4, b) R_5, c) R_6, d) R_7.

for the Heisenberg case as shown in Fig. 16. As the complexity of the problem increases, the performance of both DaTA and DaFA tends to decrease considerably. We see the same behaviour as in the other cases where DaTA learns better than DaFA.Then, we observe that the K_4 case has the lower cost function but, it did not perform as well as the others because we had a small gap in H_{target} where the optimiser got stuck. So asides for this specific K_4 case, as the problem complexity increases the overall performance decreases.

4.3.3 Comparing chip topologies

Finally, we compare the performance with different chip topologies, grid and chain. In Fig. 17 we focus on the embedding of a R_5 Heisenberg problem into a chain chip (C_5 and C_6) and we inspect the performance of DaTA after partially tracing qubits on different positions (2 and 3). We then compare the results with the case where chip and problem have the same size. Further analysis is needed but we can conclude for now that it does not matter which qubit is traced out, since as the circuit grows they all seem to converge. Moving on to the grid and chain comparison we observe slight better performance for the Ising case on K_4 and R_5 problems with DaFA for the grid (G_4 vs C_4 and G_5 vs C_5, respectively) which is a more densely connected topology than the chain. In Fig. 18 we show this same behaviour with DaTA and a harder K_4 and R_5 Heisenberg problem. We conclude by comparing G_4 and C_4 that the former has smaller deviations and has better performance. C_7 was also added into the comparison hoping than by adding extra ancilla qubits it could compete with G_4 but we do not see increase in P_0 or F_0 as we add more layers. We believe that the algorithm might be getting trapped in a local minima that is very close in energy to the true solution, as indicated by the good behaviour of the cost function. To sum up, the sparse chip topologies perform worse than denser ones. However, further work is needed to have a complete understanding of the effect of the chip topology.

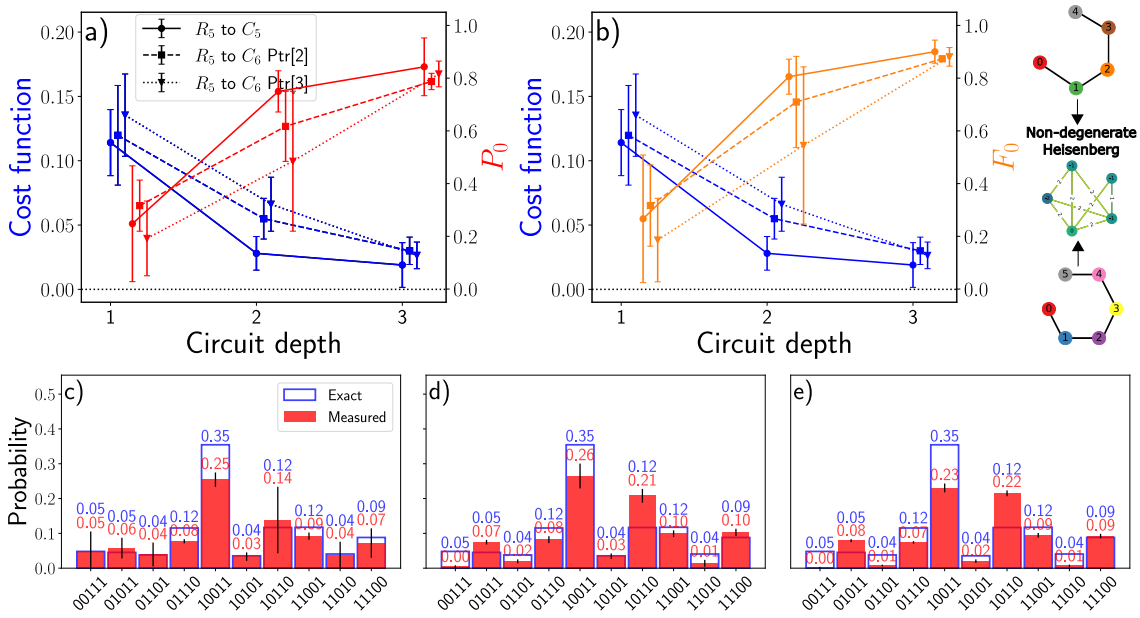

Figure 17: Chains for non-degenerate R_5 Heisenberg using DaTA a) P_0 b) F_0. Most probable 10 bitstrings overlaps for c) C_5 with (d_3, θ_{93}), d) C_6 Ptr[2] with (d_3, θ_{116}), e) C_6 Ptr[3] with (d_3, θ_{116}).

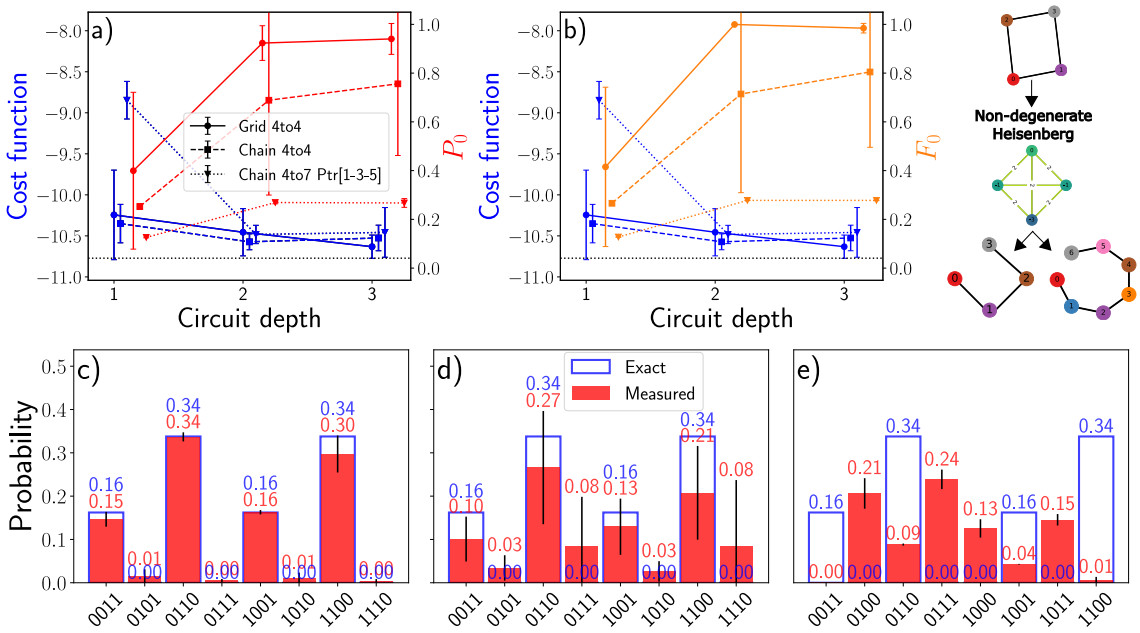

Figure 18: G_4 vs C_4 vs C_7 Ptr[1-3-5] for non-degenerate K_4 Heisenberg using DaTA a) P_0 b) F_0. Most probable 8 bitstrings overlaps for c) G_4 with (d_3, θ_{92}), d) C_4 with (d_3, θ_{70}), e) C_7 with (d_3, θ_{139}).

As for a correct comparison we should scale both the chip and problem sizes, because we can see on Fig. 19 how for small cases the grid and chain graphs can be almost equal in terms of their topology and perform similarly.

4.4 Further results

We also implemented other alternatives trying to enhance DaA. However, as this is still preliminary work and that has only been briefly studied we decided not to cover it in

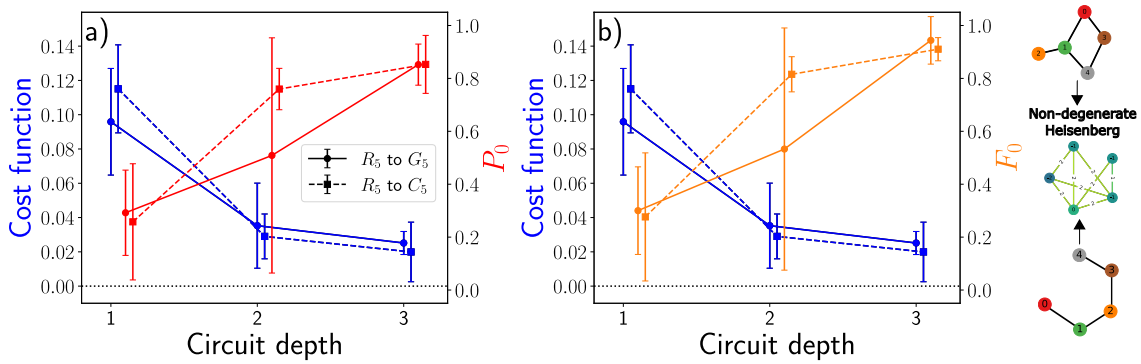

Figure 19: G_5 vs C_5 for non-degenerate R_5 Heisenberg using DaTA a) P_0 b) F_0.

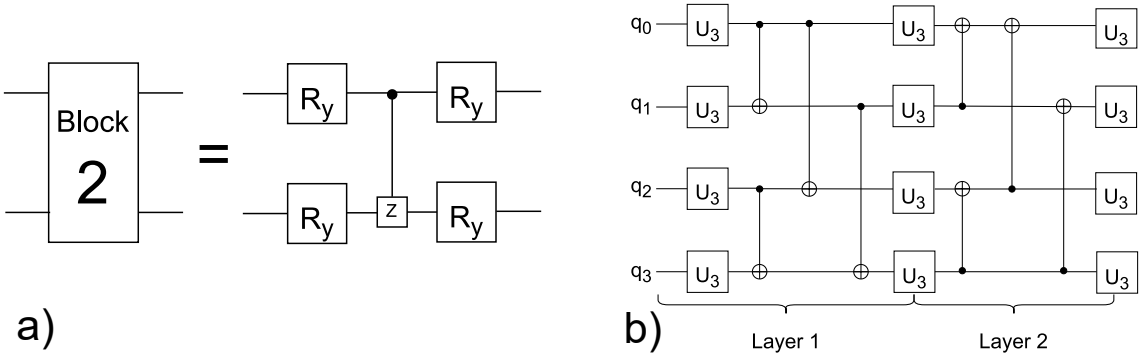

Figure 20: a) Building block composed of RY and CZ. b)Variational layers for a G_4 reduced circuit ansatz with an average of 24 parameters per layer

detail. First, we included a different cost function that only averages over a fraction of the lower energy states based on the Conditional Value at Risk (CVaR), which according to literature results is supposed to improve performance (see appendix 6.4). Second, we did a brief analysis with blocks made of parameterised single-qubit rotations in the Y direction (RY) and parameterised controlled-phase gates (CZ) as entangling gates (see appendix 6.7 or Fig. 20 a)). For each block, we only need four parameters as RY takes a single one while U3 takes three. However, we needed an extra parameter for the entangling gates to achieve a universal set of rotations made of {RY, CZ, $\mathbb{1}$} something unattainable with a fixed CZ. Third, another ansatz proposal with a greater ratio of two-qubit to single-qubit gates (see appendix 6.9) was considered, it was named reduced circuit approach and it is presented in Fig .20 b). As this procedure tends to require a higher number of layers due to the lesser number of parameters, we end up applying more two-qubit operations, which are generally more faulty and thus we decided not to use this case. However, this ansatz should be explored as it reduces the parameter count and might benefit the optimisation. Fourth, we studied a more advanced case where we use parameterised controlled U3 (CU3) entangling gates and add a penalty to the cost function depending on how far they are from being the identity. This regularisation prunes superfluous entangling gates by making them to be close to the identity and therefore decreasing the experimental error. We show some preliminary tests in the Appendix 6.8 where we used the reduced circuit as well. More work is needed on refining the parameters of the regularisation cost. Finally, during the code development we used non-noisy states, as in VQE, for testing. This could be used for our next step, compare VQE with DaA.

5 Conclusion and future work

We have presented two digital-assisted adiabatic algorithms, DaA, that combine two models of QC using variational quantum circuits and QA. This way, we build an extension of VQE that takes as an input a density operator prepared with an annealer with the aim of getting rid of the hardware limitations of the device in terms of connectivity and allowed interactions. This digital processing thus allow us to enhance the capabilities of an Ising-like annealer and reach arbitrary final states. We evaluated the performance of two different proposals: DaFA, in which the parameter optimisation entirely happens in the digital part, and DaTA, in which both digital and analogue parts get optimised. For this analysis we have assessed a set of efficiency parameters for different types, sizes and topologies of problem Hamiltonians; size and topologies of the device; inverse temperature and number of circuit layers. While further research is required for a complete assessment of these two methods, we can draw the following conclusions from the analysis presented here:

- In general,both DaTA and DaFA methods can be used to learn an arbitrary ground state and thus provide a good solution to the embedding problem. It is important to note that DaTA is more advantageous as it effectively reduces the thermal noise of our annealing process and typically requires shallower circuits than DaFA while getting higher probabilities of measuring the right solution.

- The Heisenberg and degenerate cases pose a greater challenge than Ising and non-degenerate cases but DaA is ultimately capable of solving them all provided we have enough layers on the circuit.

- Our results indicate that the required resources for the embedding of a problem with DaA are higher in terms of circuit depth for larger chips (i.e. with additional ancillary qubits). We also note that the choice of ancillary qubit to be traced out does not generally change the overall result. However, an expanded analysis on the scaling of the chip is required as with the current analysis we cannot guarantee whether too many ancillary qubits are always detrimental to performance or not.

- As we increase the size of the problem to be embedded, our preliminary results indicate a decrease in the performance of both methods. This is especially apparent for the Heisenberg case, even though a more extensive analysis is required.

- When analysing different hardware topologies, we observe that both methods have better performance when the topology is dense.

Finally, we plan to further analyze what has been mentioned in Section 4.4 along with some noisy simulations on the chip, circuit and finite number of measurements emulating the uncertainties of current quantum computers. Also, we also want to explore whether optimising the chip and circuit separately improves the overall optimisation process' speed. In addition, other parameters of the annealing process, such as the annealing schedule, could be optimised and investigated as well. Overall, we showed how DaA can be used as an alternative to heuristic-based embeddings. We therefore believe that the proposed approach of assisting annealing with variational digital processing has promising prospects for the realisation of practical quantum computations in the NISQ era and thus is an interesting venue for further research.

Bibliography

[1] Michael A. Nielsen and Isaac L. Chuang. Quantum computation and quantum information (10th anniversary edition). 2010.

[2] Moll Nikolaj et al. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology*, 3, 2018.

[3] Ari Mizel, Daniel A Lidar, and Morgan Mitchell. Simple proof of equivalence between adiabatic quantum computation and the circuit model. *Physical review letters*, 99(7):070502, 2007.

[4] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[5] Andrew M Childs, Edward Farhi, and John Preskill. Robustness of adiabatic quantum computation. *Physical Review A*, 65(1):012322, 2001.

[6] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution. *arXiv: Quantum Physics*, 2000.

[7] Tameem Albash and Daniel A Lidar. Adiabatic quantum computation. *Reviews of Modern Physics*, 90(1):015002, 2018.

[8] Andrew Lucas. Ising formulations of many np problems. *Frontiers in physics*, 2014.

[9] William M. Kaminsky, Seth Lloyd, and T. P. Orlando. Scalable superconducting architecture for adiabatic quantum computation. *arXiv: Quantum Physics*, 2004.

[10] Vicky Choi. Minor-embedding in adiabatic quantum computation: Ii. minor-universal graph design. *Quantum Information Processing*, 10(3):343–353, 2011.

[11] Jun Cai, William G. Macready, and Aidan Roy. A practical heuristic for finding graph minors. *ArXiv*, abs/1406.2741, 2014.

[12] Vicky Choi. Minor-embedding in adiabatic quantum computation: I. the parameter setting problem. *Quantum Information Processing*, 7(5):193–209, 2008.

[13] Zhengbing Bian, Fabian Chudak, Robert Brian Israel, Brad Lackey, William G Macready, and Aidan Roy. Mapping constrained optimization problems to quantum annealing with application to fault diagnosis. *Frontiers in ICT*, page 14, 2016.

[14] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, and Simon C Benjamin et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

[15] Abhinav Kandala and Antonio Mezzacapo et al. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549:242–246, 2017.

[16] Marco Cerezo, Kunal Sharma, Andrew Arrasmith, and Patrick J. Coles. Variational quantum state eigensolver. *arXiv: Quantum Physics*, 2020.

[17] A García-Sáez and JI Latorre. Addressing hard classical problems with adiabatically assisted variational quantum eigensolvers. *arXiv: Quantum Physics*, 2018.

[18] Aric Hagberg et al. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), 2008.

[19] S Efthymiou, S Ramos-Calderer, C Bravo-Prieto, A Pérez-Salinas, D García-Martín, A Garcia-Saez, JI Latorre, and S Carrazza. Qibo: a framework for quantum simulation with hardware acceleration. *Quantum Science and Technology*, 7(1):015018, dec 2021.

[20] Sukin Sim, Peter D Johnson, and Alán Aspuru-Guzik. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, 2(12):1900070, 2019.

[21] Panagiotis Kl Barkoutsos, Giacomo Nannicini, and Anton Robert et al. Improving variational quantum optimization using cvar. *Quantum*, 4:256, 2020.

[22] Pablo Díez-Valle, Diego Porras, and Juan José García-Ripoll. Quantum variational optimization: The role of entanglement and problem hardness. *Physical Review A*, 104(6):062426, 2021.

6 Appendix

6.1 QUBO to Ising

We need to transform QUBO problems' variables $x \in \mathbb{B}^N (N \in \mathbb{N})$ with coefficients $Q_{ij} \in \mathbb{R}$ for $1 \leq j \leq i \leq n$ into the Ising model, which requires Ising spin variables $z_i \in \{-1, +1\}$. The requires applying $x_i \Rightarrow \frac{1-z_i}{2}$, which yields:

$$\min_x [\sum_i^n Q_i x_i + \sum_{i<j} Q_{i,j} x_i x_j] = \min_z \sum_i^n Q_i \frac{1-z_i}{2} + \sum_{i<j} Q_{i,j} \frac{1-z_i}{2} \frac{1-z_j}{2} = \tag{11}$$

$$\min_z - \sum_i^n Q_i \frac{z_i}{2} - \underbrace{\sum_{i<j} Q_{i,j} \frac{z_i}{4} - \sum_{i<j} Q_{i,j} \frac{z_j}{4}}_{\sum_j Q_{i,j} \frac{z_i}{2}} + \sum_{i<j} Q_{i,j} \frac{z_i z_j}{4} + \underbrace{\sum_i^n Q_i \frac{1}{2}}_{constant} + \sum_{i<j} Q_{i,j} \frac{z_j}{4} = \tag{12}$$

$$\min_z \left\{ - \sum_i^n \frac{1}{2} (Q_i + \sum_j Q_{i,j}) z_i + \underbrace{\sum_{i<j} J_{i,j} z_i z_j}_{J_{ij} = \frac{Q_{ij}}{4}} \right\} = \arg\min_{z \in \{1,-1\}^n} \sum_i h_i z_i + \sum_{i<j} J_{ij} z_i z_j \tag{13}$$

We remind the reader that constants can be neglected as they do not change the minimum. Both formulations are equivalent via the bijective relations: $h_i = -\frac{1}{2}(a_i + \sum_j b_{ij})$, $J_{ij} = \frac{b_{ij}}{4}$.

6.2 Basic graph theory

We use graphs all throughout this work. Specifically, undirected graphs with weighted nodes and edges. To understand what this implies, we made this brief introduction to graph theory for undirected simple graphs (see an example graph of this kind in Fig. 21).

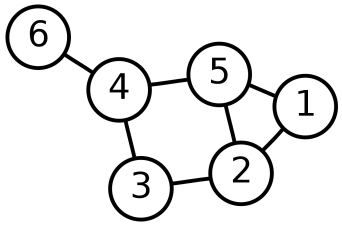

Figure 21: Undirected simple graph with 6 nodes and 7 edges

- **Graphs**: Mathematical structures used to model pairwise relations between objects. A graph G is an ordered $G = (N, E)$ comprising: N, a set of nodes; and $E \subseteq \{\{x, y\} \mid x, y \in V \text{ and } x \neq y\}$ a set of edges, each one associated with two distinct nodes.
- **Undirected simple graphs**: The edges link two nodes symmetrically. Multiple edges, i.e. two or more edges that join the same two nodes, or loops, i.e. when an edge starting and ending on the same vertex, are not allowed.
- **Graph order** $|N|$: The graph's number of nodes.
- **Graph size** $|E|$: The graph's number of edges.

- **Vertex degree**: Number of edges that are incident to a vertex.
- **Graph degree**: Maximum of the degrees of its nodes.
- **Regular graph**: Graph in which every vertex has the same degree.
- **Complete graph** K_n: Regular graph of order n where all nodes have the maximum degree, $n-1$.
- **Graph minor**: An undirected graph D is called a minor of the graph G if D can be formed from G by deleting edges and nodes and by contracting edges.
- **Planar graph**: A graph that can be drawn without crossings on the plane.
- **Treewidth**: Informally, an integer specifying, how far the graph is from being a tree. Used in graph algorithms analysis for their parametrised complexity. Many NP-hard algorithms for general graphs become easier when the treewidth is bounded by a constant.
- **Graph automorphism**: Form of symmetry in which the graph is mapped onto itself while preserving the edge–node connectivity. Formally, an automorphism of a graph $G = (N, E)$ is a permutation σ of the node set N, such that the pair of nodes (u, v) form an edge if and only if the pair (σ(u),σ(v)) also forms an edge.

6.3 Dwave's embedding algorithm

6.3.1 Graph problem

Embedding a problem graph $G = (N_G, E_G)$ into a hardware graph $D = (N_D, E_D)$ requires mapping φ of each node in N_G to a subset of nodes in N_D:

$$\varphi : N_G \to 2^{N_D},$$

where 2^{N_D} is the set of all subsets of N_D, that fulfils:

1. For each node v in N_G, the set of nodes $\varphi(v)$ induces a connected subgraph in D, called the chain of v. Chains can be composed single qubits.
2. For every edge $e = \{u, v\} \in E_G$ there exist nodes $\tilde{u} \in \varphi(u)$ and $\tilde{v} \in \varphi(v)$ such that $\{\tilde{u}, \tilde{v}\} \in E_D$. If u and v are connected on D they would also be in $\varphi(x)$ and $\varphi(y)$.
3. If $\varphi(v) \cap \varphi(u) = \emptyset$ for all $u \neq v \in N_G$, i.e., each node $\tilde{v}$ of D appears in the mapping of at most one node of G, $\phi(x)$ and $\phi(y)$ are disjoint if $x \neq y$. In other words, chains do not share logical qubits.

Only if all three conditions are satisfied a minor-embedding will be achieved. To find $\varphi : N_G \to 2^{N_D}$ the algorithm would construct the node model of G in D and iteratively improve the embedding by examining each node $x \in N(G)$ and rebuilding its chain in D. An improvement is defined as a reduction of the largest amount of chains using a determined qubit. Otherwise, cutting down the quantity of qubits used would be the secondary goal.

6.3.2 Parameter setting

The chain parameters need to be set carefully in order to keep its qubits with the same binary value during the anneal without distorting the problem. New parameters must solve the original problem on G by solving the embedded one on D. A one-to-one correspondence between the minimums of $\mathcal{E}$ and $\mathcal{E}^{emb}$ is sought. Intuitively, they maintain h_{ij} and J_{ij} for any element not belonging to any chain of length C_L. We did a study on all this on 6.3.3 but first we show our parameter setting scheme for the ones chained:

1. Split h_i evenly among qubits in the logical chain, $h_i^T = \frac{h_i}{C_L}$.
2. Select a strong negative coupler C_{ij} value for all chain edges. Its absolute value must be bigger than those of regular couplings around it (i.e. $C_{ij} = -J_{ij} - 1$).
3. Compensate C_{ij} effect by adding $h_i^c = h_i^T + \frac{|C_{ij}|}{C_L}$ to each qubit $\in T_i$.

6.3.3 Minor embedding analysis

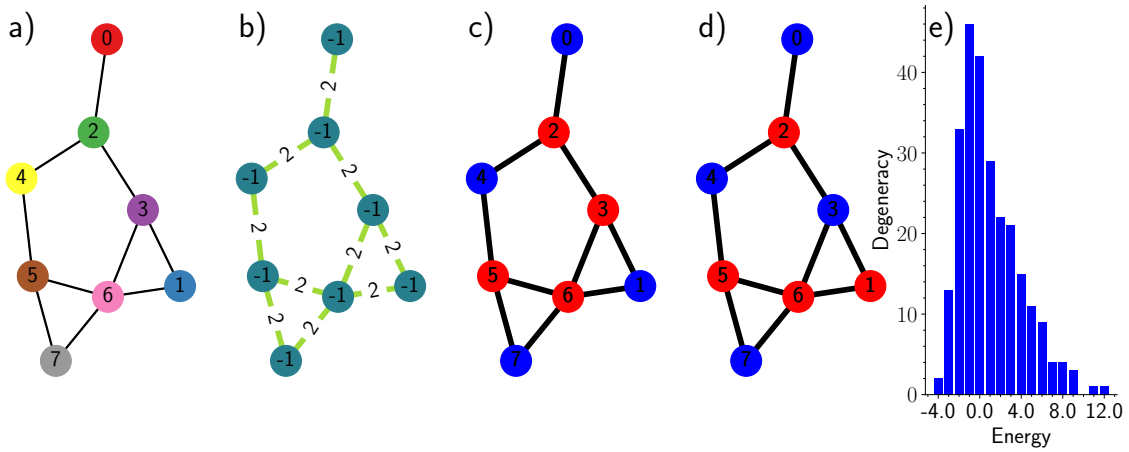

Figure 22: a) Original graph to be embedded, b) Weights, c), d) Solutions, e) Energy spectrum

We analyse their algorithm for the Maximal Independent Set (MIS) problem (See appendix 6.6). We use DWave's implementation available openly on GitHub. They keep both the energy and degeneracy of the original ground state. The most notable difference is the expected change in degeneracy for energies above the ground state for the embedded case. Let us show this with the small example of Fig. 22 a) with our parameters:

$$\left.\begin{array}{ll} h_{ij} = -1; & J_{ij} = 2 \\ h_{c_{ji}} = 1; & C_{ij} = -3 \end{array}\right\}$$

Its solutions are [0,1,4,7] and [0,3,4,7]. By using the 1 to mean it belongs to the MIS while 0 means they do not, and we build a bistring to name each state. So, those solutions as a bistring would be [11001001] and [10011001], respectively. If you look at the coloured solutions on the graphs, blue nodes belong to the MIS while red ones do not. Their energy and ground state can be calculated, in this case is -4. Considering the two extra qubits required for the embedding and the new labelling, our solution would be [1100100001] and [1001100001]. In Fig. 22 we show the graph G to be embedded, its different weights, solutions and energy spectrum as a histogram where to acknowledge the degeneracy of each state. In Fig. 23 we show more similar graphs containing the optimal embedding alongside

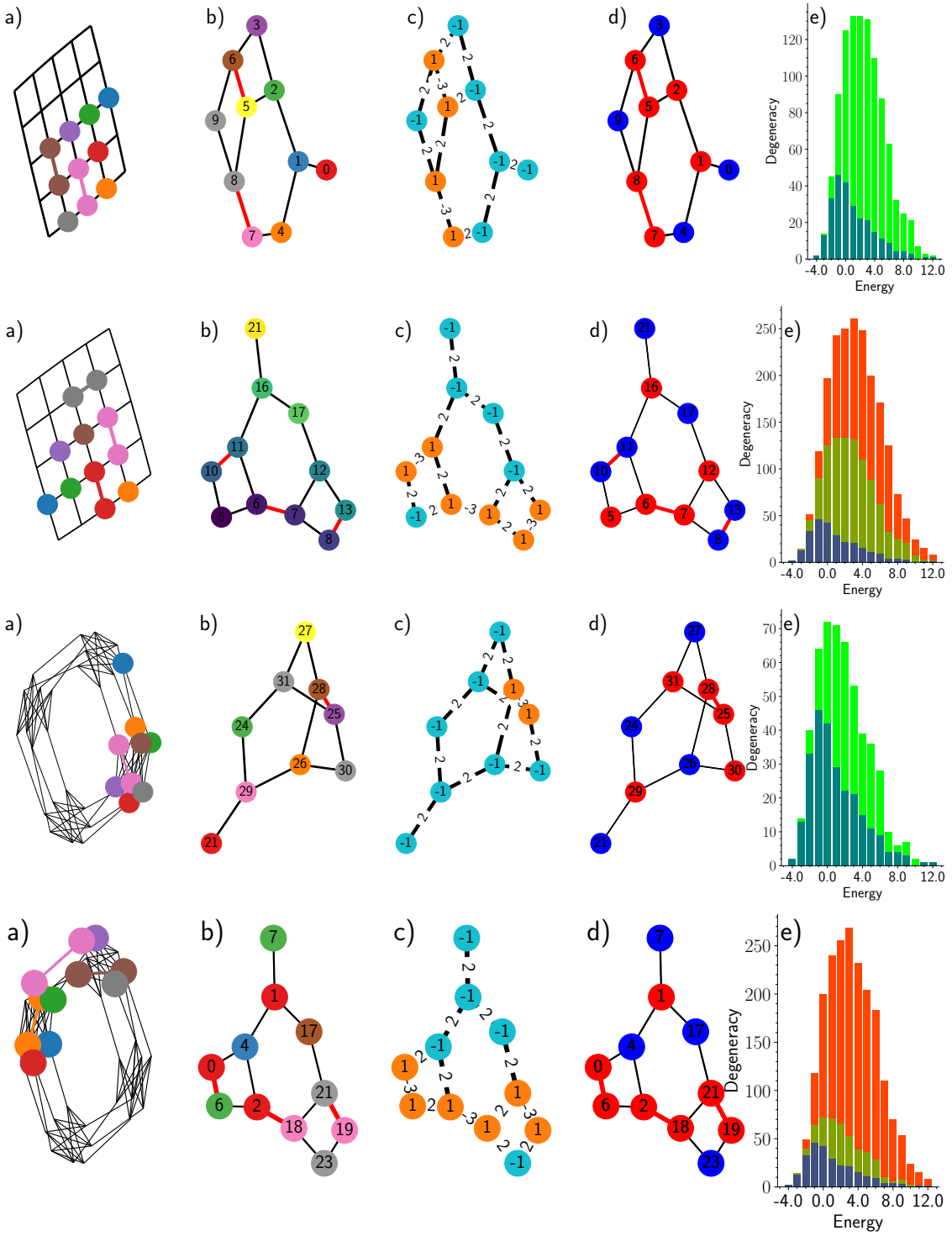

Figure 23: Top: Optimal minor embedding on a 25 element 1^{st} nearest neighbours grid or G_{25}. Mid-Top: Random minor embedding on G_{25}. Mid-Bottom: Optimal minor embedding on 2x2 Chimera. Bottom: Random minor embedding on 2x2 Chimera. a) Minor embedding, b) Labels, c) Weights, d) One of the two solutions, e) Energy Spectrum.

another random one generated by Dwave's algorithm for the grid and chimera hardware graphs. To them we superimpose different colours for the original problem graph and optimal embedding energy spectra for a more insightful comparison. Blue for the original

graph, green for the optimal embedding and red for a random embedding different from the optimal one.

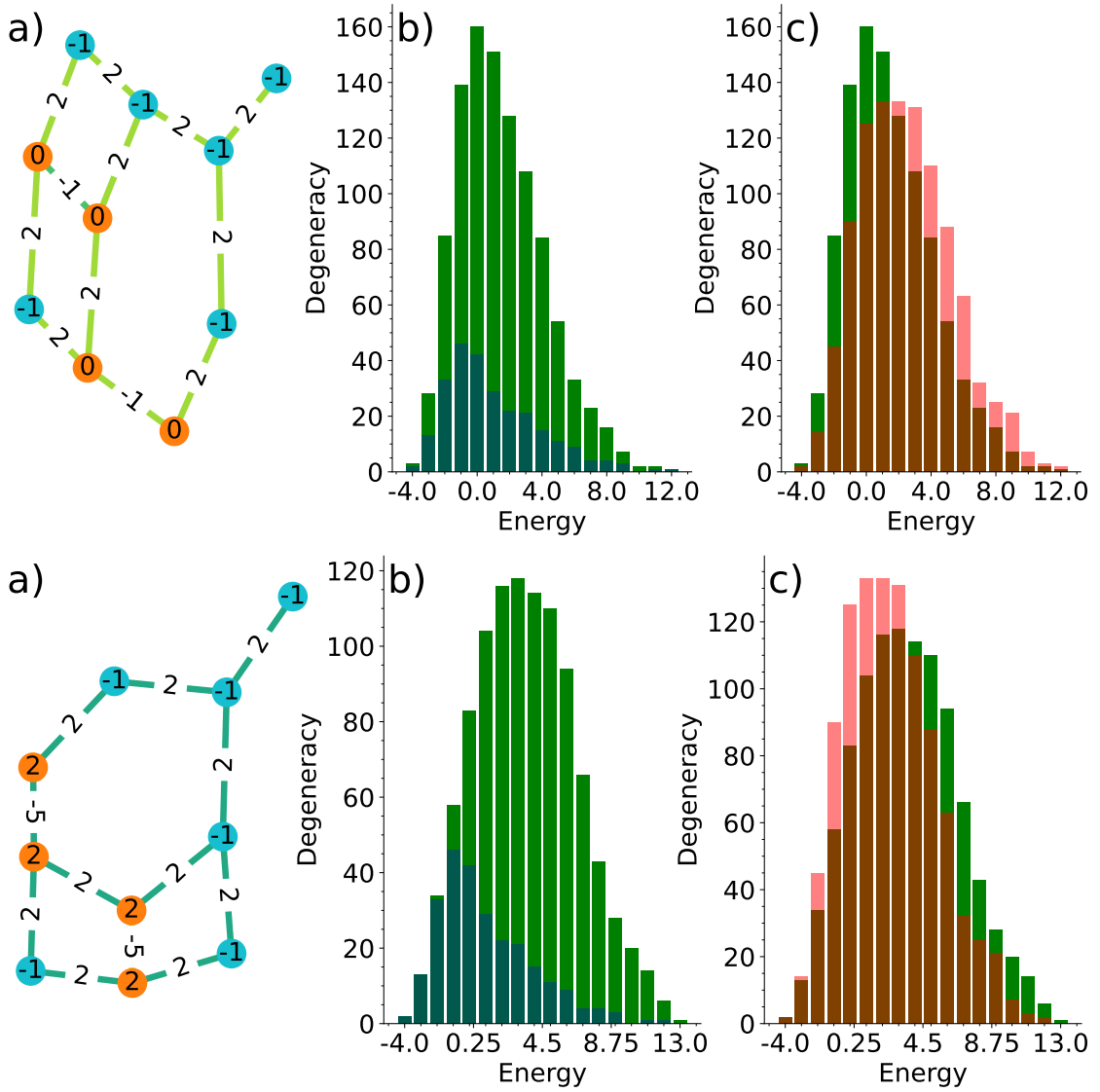

Figure 24: Top: $C^*_{ij} = -1$, Bottom: $C^*_{ij} = -5$. a)Optimal embedding weights with the given C^*_{ij}, b) Energy spectrums of the original problem and the embedding with the given C^*_{ij}, c) Energy spectrums of the embeddings with $C_{ij} = -3$ and the given C^*_{ij}.

We justify our choice of $|C_{ij}|$ to be slightly greater than J_{ij}, $C_{ij} = -3$. For this, we will compare the different energy spectrums generated by varying C_{ij} for the optimal embedding on the grid topology with the original and $C_{ij} = -3$ energy spectrums in Fig. 24 and Fig. 25. In those figures blue represents the original graph, green the embedded graph with the given C^*_{ij} and red the embedded graph with $C_{ij} = -3$. In summary, we see how the energy spectrum shifts to higher energies as we increase $|C_{ij}|$ until it reaches a point where setting it too high distorts the problem. However, we can see how setting the chain strength too low compared to the problem's biases leads to the appearance of more states whose energy is the same as the ground state, ones which could wrongfully be labelled as solutions.

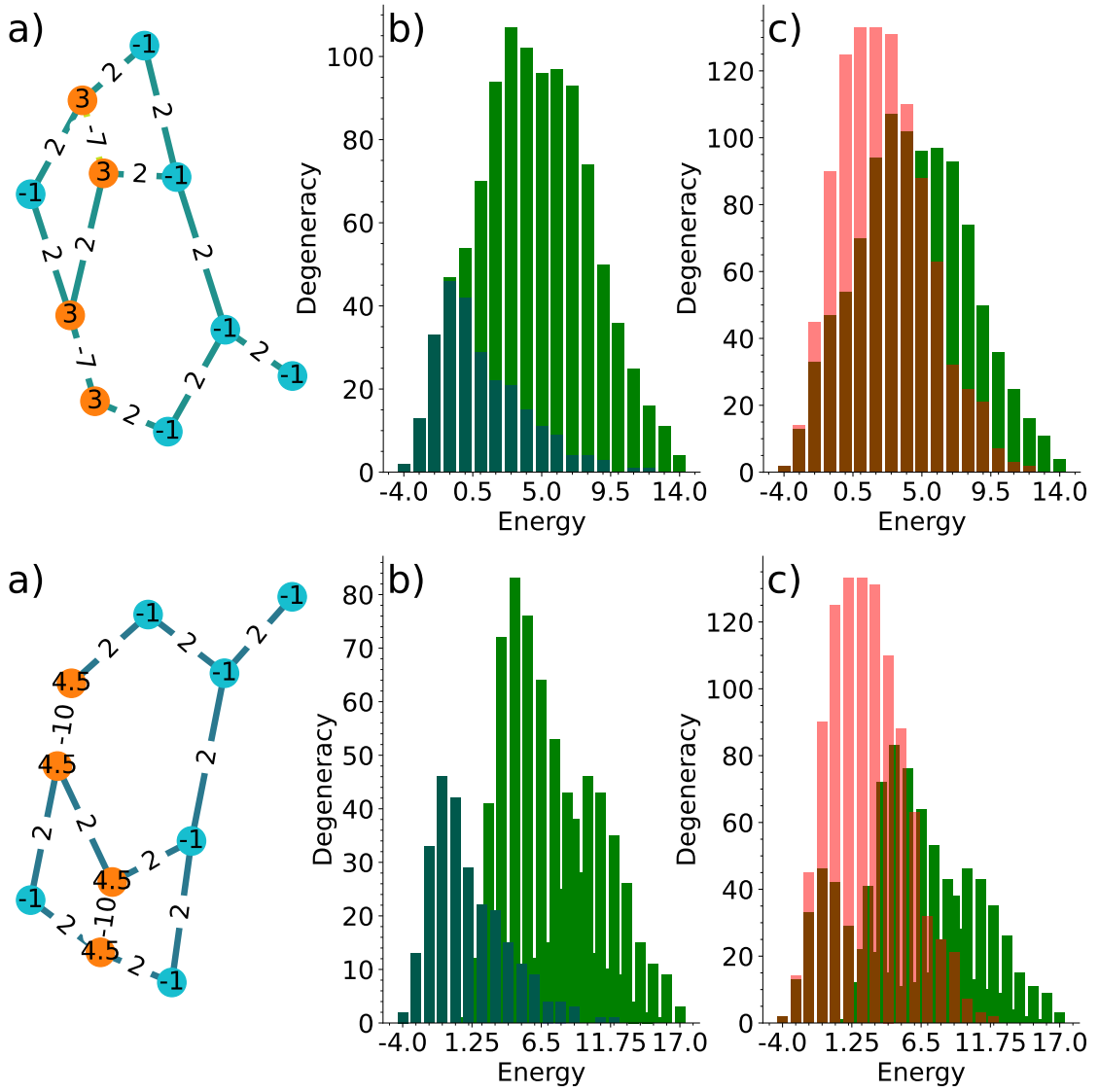

Figure 25: Top: $C^*_{ij} = -7$, Bottom: $C^*_{ij} = -10$. a)Optimal embedding weights with the given C^*_{ij}, b) Energy spectrums of the original problem and the embedding with the given C^*_{ij}, c) Energy spectrums of the embeddings with $C_{ij} = -3$ and the given C^*_{ij}.

6.3.4 Limitations

Here we just show that Minorminer is unable to embed small graphs on a 1^{st} nearest neighbours grid in Fig. 26. But it is capable of finding embedding of more complex G on other graphs that may be even more limited, like on a random graph of degree 3 in Fig. 27.

6.4 CVAR cost function

This cost function envisioned by [21] only works with a fraction $\alpha \in \{0, 1\}$ of the lower energy states to compute its value unlike the general cost function that consisted in an average over all states. It is called Conditional Value at Risk (CVaR) cost function as it is inspired by this approach. We can calculate it as follows, being λ_k the eigenvalues sorted

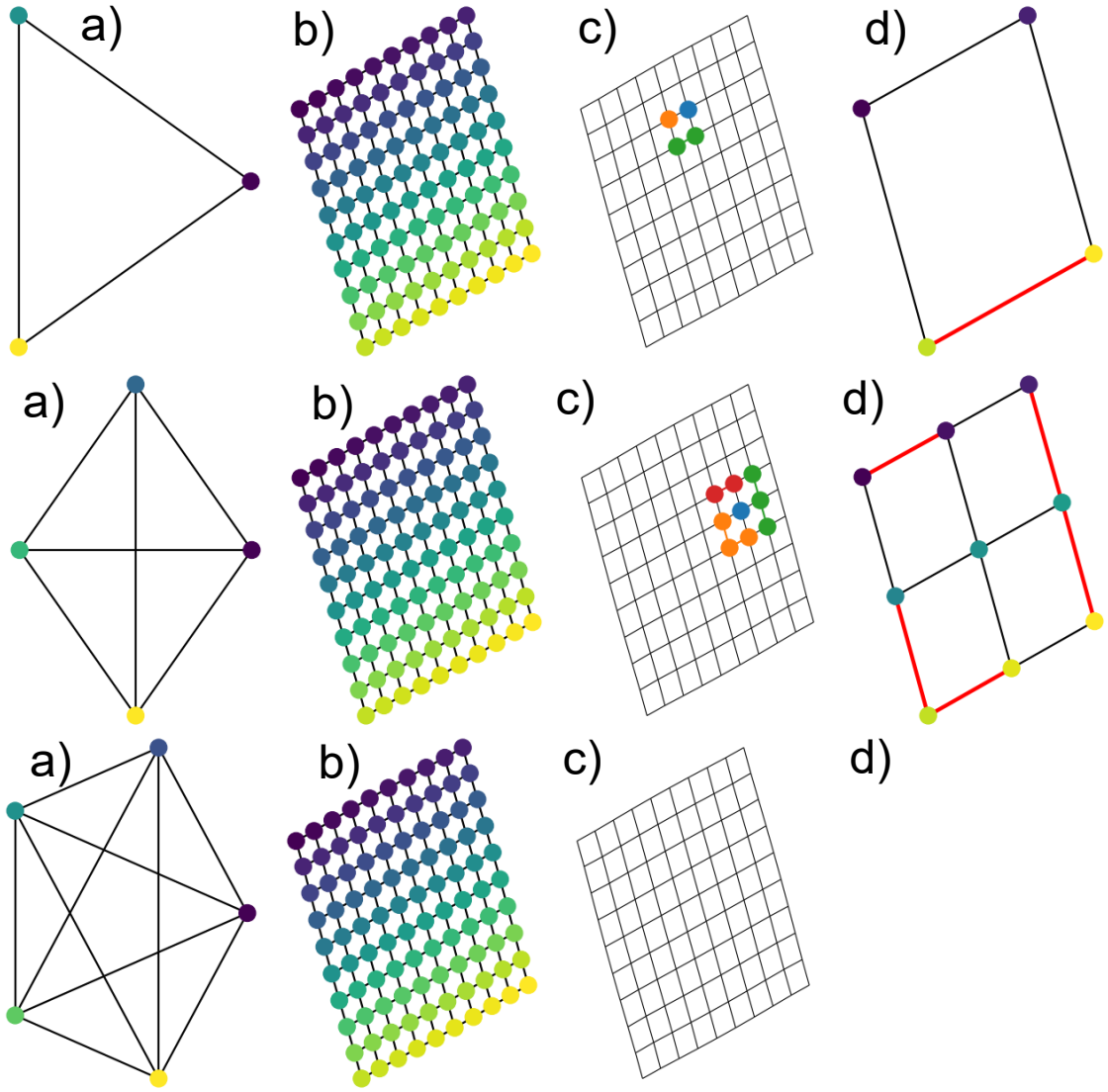

Figure 26: Top: K_3 on a 100 element 1^{st} nearest neighbours grid or G_{100}, Mid: K_4 on G_{100}, Bottom: K_5 could not be embedded even on G_{10000}, here we just show the already packed G_{100}. a)Problem Graph, b) Hardware graph, c)Minor embedding, d) Embedded graph.

in non decreasing order of our final state.

$$CVaR_\alpha = \frac{K}{\lceil \alpha 2^n \rceil} \sum_{k=0}^{\lceil \alpha 2^n \rceil} \lambda_k \tag{14}$$

$X(\theta)$ is a random variable composed the eigenvectors of Hamiltonian. The CVaR cost is the expected value of the lower α-tail of the X distribution.

$$X(\theta) = H_{j,j} for j \in \{0,1\}^n \tag{15}$$

$$Prob(X(\theta)) = H_{j,j} = \mathrm{Tr}[\mathrm{Tr}_A[U(\theta)\rho_{device}U^\dagger(\theta)]|j\rangle\langle j|] \tag{16}$$

Note that in the limits $\alpha \to 0$, $CVaR_{\alpha\to 0} = \lambda_0$ and $\alpha \to 1$, $CVaR_1$ corresponds to the regular average or expected value of the energy we used on the main work. So, CVaR

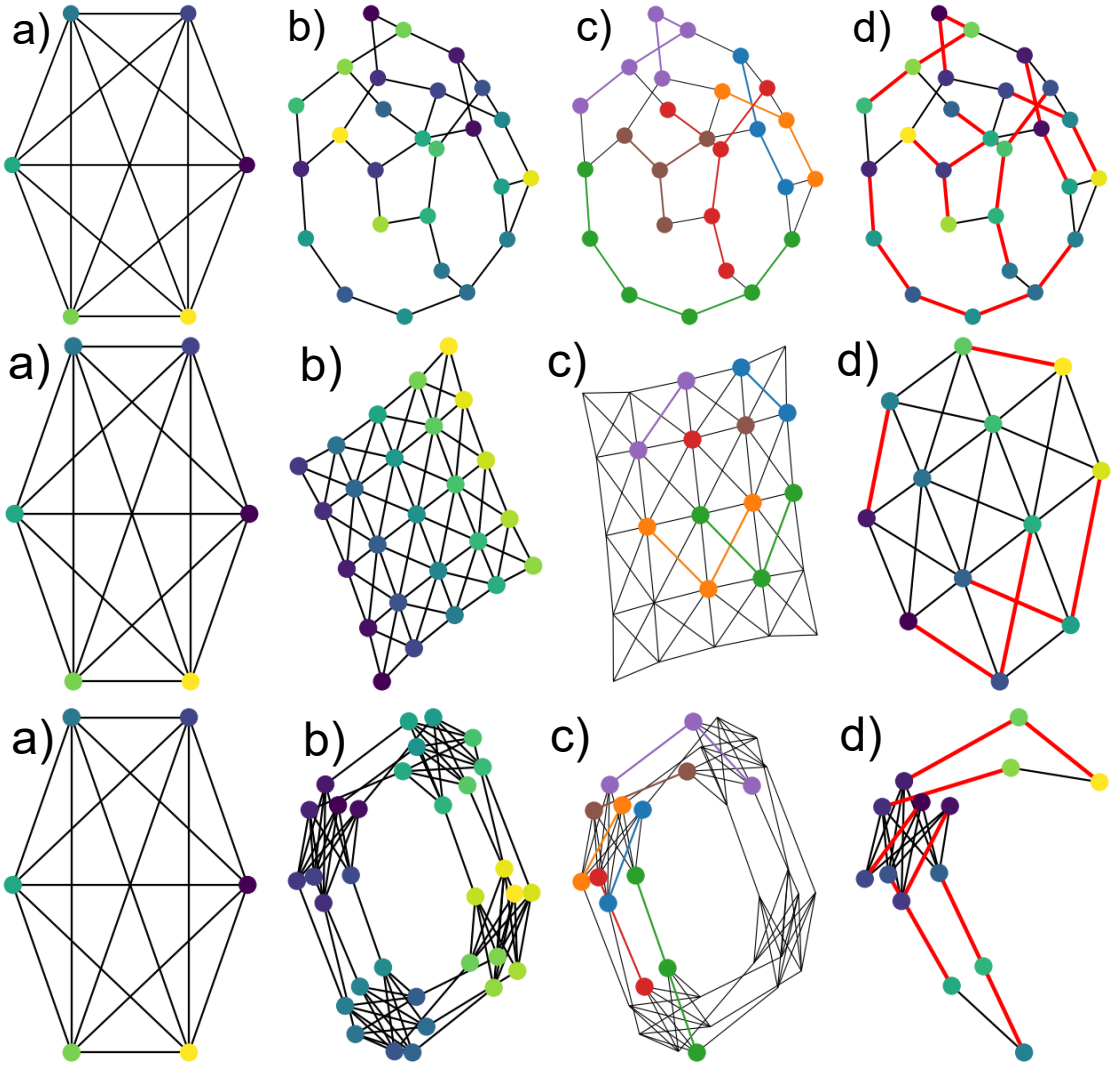

Figure 27: Top: K_6 on random 30 node 3 degree per node graph, Mid: K_6 on 10x10 1^{st} and 2^{nd} nearest neighbours connected grid, Bottom: K_6 on 2x2 Chimera cell graph. a)Problem Graph, b) Hardware graph, c)Minor embedding, d) Embedded graph.

mixes an expected value and a minimum. Hence, we are interested in small α values regarding only the lower energy states but by having several of them instead of just λ_0 so the optimiser handles the optimisation process in a easier way. All in all, we want to explore the claimed improvements on success rate and speed for any problems for DaA and how changing α may affect its performance. A prototype has already been coded and tested but, at the moment, $CVaR_1$, takes longer to compute than the cost function used on the main work although they are equivalent. This was due to the lack of optimisation in the code, an issue to be resolved in the future. Anyway, for tested $CVaR_{0.1}$ and $CVaR_{0.2}$ we saw slight time improvements over $CVaR_1$ without affecting the quality of the results. For DaFA the overlap with the ground state improved slightly and for DaTA the overlap could not be improving as it reached almost the unity in all cases. Further analysis is needed for bigger systems, where this method is expected to outperform the previous one by cutting a greater number of states from the cost function.

6.5 Classical optimiser choice

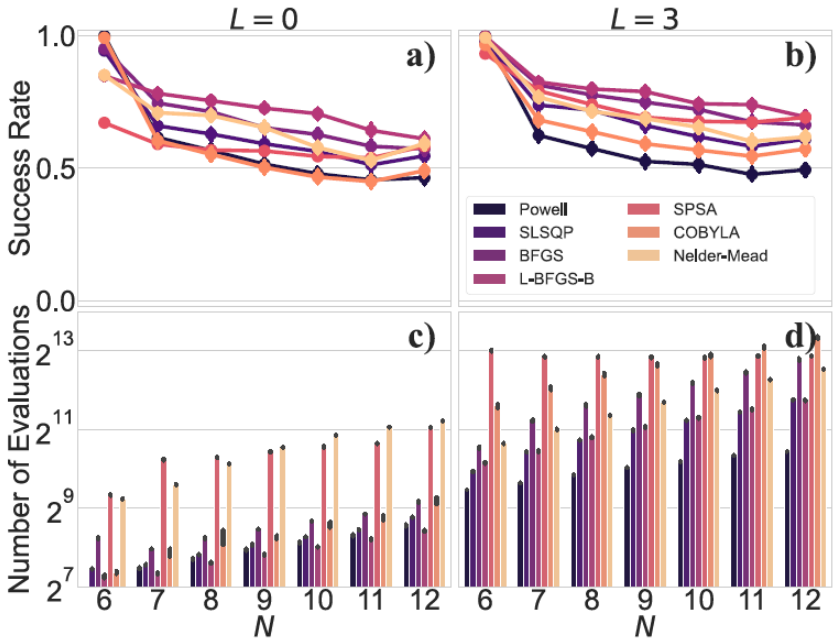

Figure 28: From [22], comparison of VQE performance with a variety of classical optimisation methods to optimise the variational parameters using exact quantum states resulting from simulation. From left to right, they increase the number of layers L of the ansatz. On the X axis, they plot the size of the problem: (a),(b) success rate, (c),(d) objective function evaluations needed to converge. The results indicate the average of 1600 instances and a 95% confidence interval. The better would be the better in terms of higher chance of success and lower function evaluations.

First of all, we use the optimisation methods from QUBO that come from those provided by SciPy. Note that all of them could converge to a local minimum, since classical optimisers use convergence criteria based on the change of the cost function in each step. Therefore, convergence of the algorithm does not imply success. Fig. 28 from [22] shows different performance depending on the optimiser used. They also concluded that gradient-free optimisers such as SPSA, COBYLA, Powell, and Nelder-Mead perform well even when the information of the objective function is not complete. Whereas, Gradient-based optimisers such as SLSQP, BFGS, and L-BFGS-B perform very well in wave function simulations. However, they fail in the last case, as the descent direction is not computed properly. As our case was more similar to the last we choose the best performing one for our code. Hence, we use L-BFGS-B for all this thesis.

6.6 WMIS and MIS problem

The classical optimisation problem we study for this work will be the Maximal Independent Set (MIS) problem, an example of which can be found in Fig. 29. A MIS is a set of nodes of which no pair of them are adjacent, that is not a subset of any other independent set. It is a special case of the Weighted Maximal Independent Set (WMIS) problem where $h_i = h$ for all the nodes. For WMIS, node weight has to be considered in a way that smaller independent sets may be preferable, for their accumulated weight, rather than bigger ones with less value overall. This problem is described by the following Hamiltonian:

$$H^{WMIS} = \sum_{i \in V} h_i z_i + \sum_{i,j \in E} J_{ij} z_i z_j \qquad \text{with} \quad z_i = (1 - \sigma_i^z)/2 \tag{17}$$

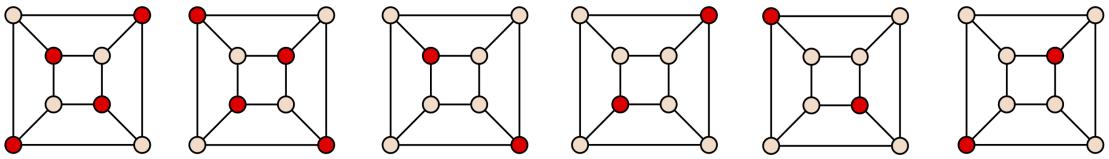

Figure 29: Edited from https://commons.wikimedia.org/wiki/File:Cube-maximal-independence.svg. This graph has six different maximal independent sets (the first two are maximum), shown as the red nodes.

Whose eigenvalues are, as calculated in [12]:

$$\mathcal{E}\left(z_1, \ldots, z_n\right) = -\sum_{i \in \mathrm{V}(G)} h_i z_i + \sum_{ij \in \mathrm{E}(G)} J_{ij} z_i z_j \tag{18}$$

6.7 RY-CZ gate set

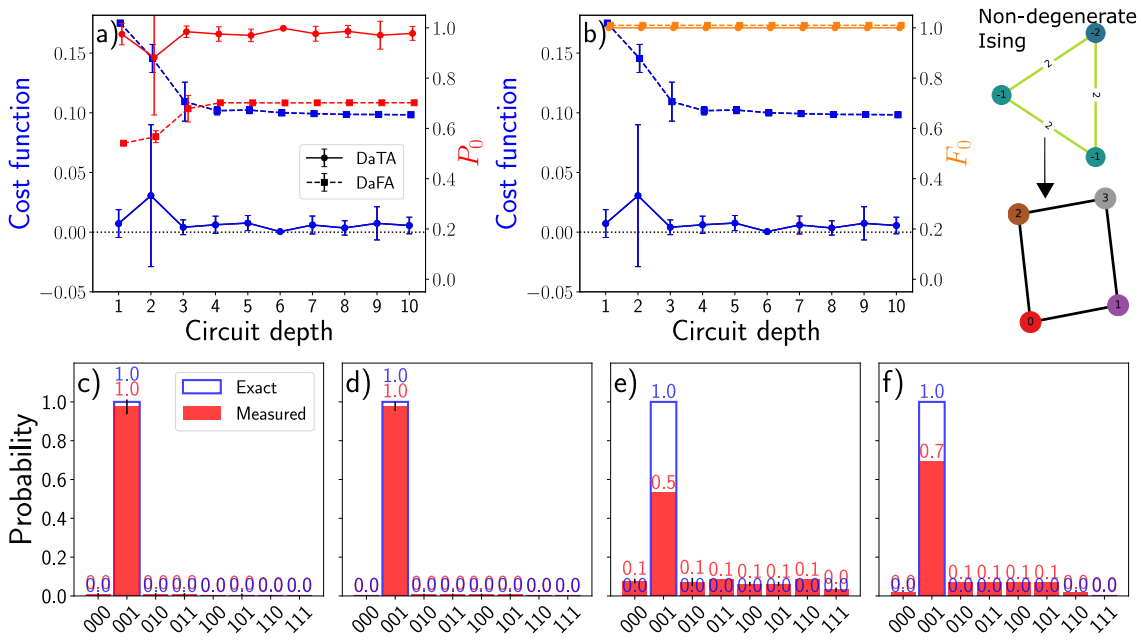

Figure 30: DaTA vs DaFA for non-degenerate K_3 Ising into G_4 with the reduced block ansatz with RY-CZ gates. a) P_0 b) F_0. All bitstrings overlaps for c) DaTA with (d_1, θ_{24}), d) DaTA with (d_3, θ_{48}), e) DaFA with (d_1, θ_{16}), f) DaFA with (d_3, θ_{40}).

First we note that Y,Z rotations are universal, so in this case our entangling gate CZ needs to be parameterised and we need to change control-target qubits each layer to implement full Z rotations on each qubit. That way we have an universal circuit with the block in Fig. 20 a). Another option would be to add a single-qubit Z rotation in front of every RY gate but this makes the circuit more complex. Now that the gates are determined we can say this approach is lighter on the number of parameters used than U3-CNOT. Because, single-qubit rotations are now implemented using a RY gate which

only uses a single parameters instead of the three used by a U3 gate.

$$RY(\theta) = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix} \qquad CZ(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & e^{-i\phi/2} & 0 \\ 0 & 0 & 0 & e^{i\phi/2} \end{pmatrix} \tag{19}$$

For our small conducted test in Fig. 30 we observe how we require more layers to reach similar result than those of the U3-CNOT case. However, the RY-CZ set is way faster due to reduced parameter count. For reaching the same results while being shallower, we decided to use the circuit composed of U3-CNOT as it relies on a smaller number of two-qubit gates by having harder single-qubit gates, as the latter are more robust with the current technology. As part of the future work we would want to benchmark this gate set to analyse how the performance of the circuit changes by using this type of layers with less parameters with bigger problems and chips.

6.8 Regularised DaA

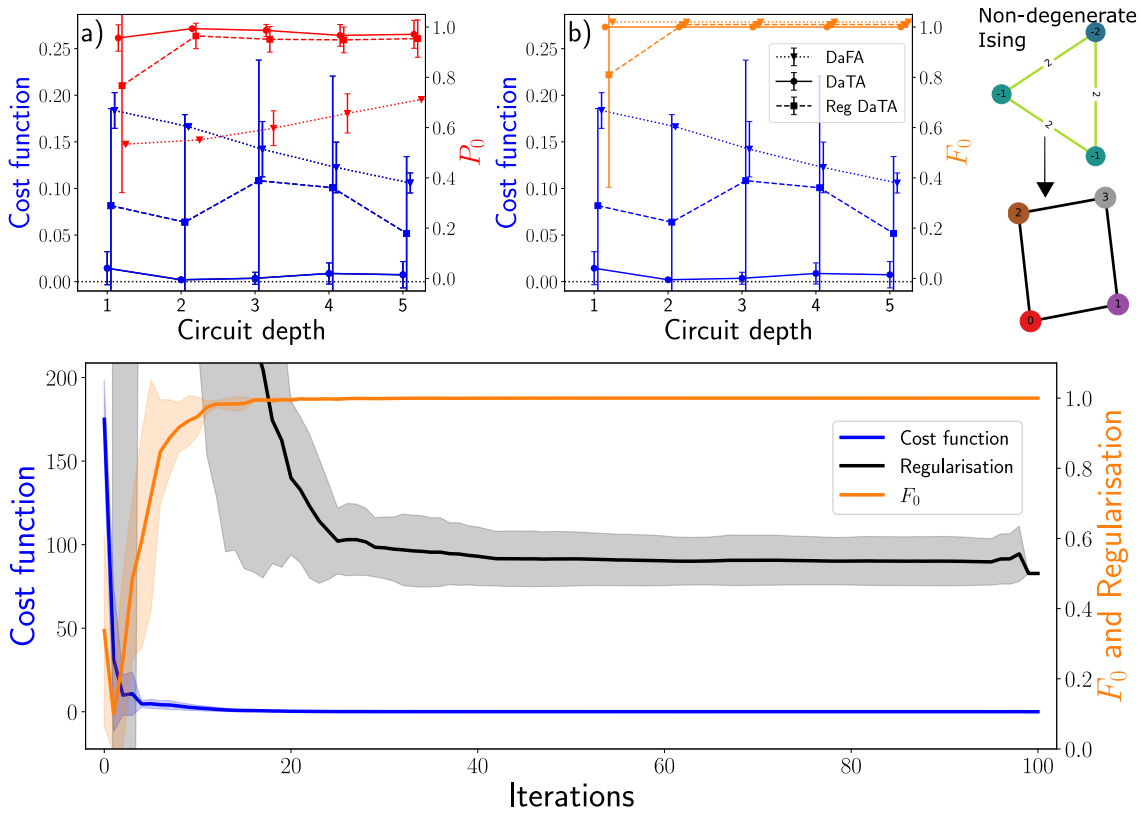

Figure 31: DaFA vs DaTA vs Regularised DaTA for non-degenerate K_3 Ising into G_4 with the reduced circuit ansatz with U3-CU3 gates. a) P_0 b) F_0. c) Shows the normalised regularised cost function. The norm used was the one corresponding to the non-regularised case to show the regularisation effect on the cost function. The cost function starts over the unity due to regularisation term which dominates over the energy term, which is bound $\in [0, 1]$, until the two-qubit gates get optimised. Afterwards, the energy term gets optimised. The regularisation penalty is plotted on the right axis. It keeps a near zero value at the end of the optimisation indicating that the entangling gates were heavily pruned without affecting our ability to reach the solution.

Another topic worth exploring is the case were all gates are tunable and the cost function has a penalty term related to how far are our entangling gates from the identity. The aim is to reduce as much as possible the two-qubit gate parameters because we lessen the experimental noise on the process not only by removing 2-qubit gates altogether, but also by making them operations closer to the identity. Hence, we prune the two-qubit gates and only consider applying them when there is a significant improvement on the energy term. The question would be what we will call "significant" as you need some kind of parameter before the two-qubit gates renormalization term for it to be of comparable magnitude to the energy while avoiding the system of over optimising the regularising term. We came up with a constant of $0.01 * |\lambda_0|$. It would also be preferred to add some kind of depth consideration to take into account the greater number of parameters as the circuit grows. A test was conducted for the reduced circuit topology with U3-CU3 gates with a G_4 chip and non-degenerate K_3 Ising problem. For the results in Fig 31, we check how the regularising term decreases in the early set of iterations indicating how we managed to turn most two-qubit gates into the identity for the simple case considered where the solution was reached practically by just using single-qubit rotations. Obviously, we do not have an improvement over DaTA as we are reducing the expressibility of our circuit but the performance is close to it, which was our initial goal. Further analysis with bigger circuits is needed to get proper conclusion of the effect of the regularising term.

6.9 Circuit resources

This is a gate and parameter count for the two different considered ansatzes, reduced blocks (Fig. 2 b)) and reduced circuit (Fig. 20 b)) for the U3-CNOT gate set. We recall how each hardware graph $D(N_D, E_D)$ was composed of a different number of nodes and edges, we will name $|N_D|$ and $|E_D|$ the node and edge count respectively. Finally, G_p are the required parameters for our parameterised gate of choice, for U3 $G_p = 3$, and d_i are the amount of circuit layers we use. Take into account, that grid and chain topologies have a different growth for $|E_D|$ depending on $|N_D|$ as the grid is a more connected topology.

- Reduced Blocks

$$\text{DaFA} = 2|E_H|G_p d_i + |N_H|G_p \qquad \text{DaTA} = \text{DaFA} + (|E_H| + |N_H|) \tag{20}$$

$$1\text{Q} = 2|E_H||N_H|d_i \qquad 2\text{Q} = |E_H|d_i \qquad 1\text{Q}/2\text{Q} = 2|N_H| \tag{21}$$

- Reduced Circuit

$$\text{DaFA} = |N_H|G_p(d_i + 1) \qquad \text{DaTA} = \text{DaFA} + (|E_H| + |N_H|) \tag{22}$$

$$1\text{Q} = |N_H|(d_i + 1) \qquad 2\text{Q} = |E_H|d_i \qquad 1\text{Q}/2\text{Q} = \frac{|N_H|(d_i + 1)}{|E_H|d_i} \tag{23}$$

The result to be drawn is that the ratio 1Q/2Q, of single-qubit gates over two-qubit gates, is always bigger for the reduced blocks case for our chips topologies considered, grid and chain. Hence, that was the one chosen for leveraging more importance on single-qubit gates.