A Quantum Annealing Approach to Graph Partitioning

Author: Andrea Pérez Martín
*Facultat de Física, Universitat de Barcelona, Diagonal 645, 08028 Barcelona, Spain.**

Advisor: Bruno Juliá Díaz

Abstract: We provide an introduction to quantum annealing and its application to graph partitioning. Simulations of small problems are performed, as well as an implementation in an actual quantum annealer through the idea of minor-embedding. The quality of the annealing results is examined by comparison to a current classical state-of-the-art method.

I. INTRODUCTION

Across multiple disciplines, graphs are frequently used as abstractions when modeling problems. Even if the final application concerns other methods, it is often useful to partition large graphs into smaller subgraphs in order to reduce graph complexity or allow parallel computing. Graph partitioning (GP) methods arose to enable this in applications such as bioinformatics, social networks, VLSI design, transportation networks, and image processing [1].

Proved to be an NP-hard problem, a wide spectrum of GP algorithms has been developed over the years. The possibility of reducing the problem to that of finding the ground state of a system of interacting spins [2] has been studied in recent years. Quantum annealing (QA) has proved to be an effective method to find the solution when the dimension of the system is large and direct computation is no longer an option.

D-Wave Systems [3] launched in 2011 the first commercial quantum annealer to the market and, since then, QA has been growing in importance. Though still far from becoming a practical technology, results are promising and QA is expected to bring many valuable applications to society in fields such as finance, life sciences and communications.

In this work, we focus on understanding how QA works implementing it to solve the GP problem. We begin in section II by defining the balanced GP problem. Then, in section III, we introduce adiabatic quantum computation (AQC) and its heuristic implementation QA and explain the Ising and QUBO models. We additionally formulate a reduced version of the GP problem using the QUBO formulation and perform a numerical simulation to solve a particular small problem. Section IV introduces D-Wave's annealers and explains how problems are implemented using the example of the previous simulation. It also explores GP problems for which a classical simulation is no longer possible and evaluates the quality of the solutions by comparison with a multilevel GP classical software. Finally, in section V we summarize the results and give the conclusions of this work.

*Electronic address: andreaperezmartin@gmail.com

II. GRAPH PARTITIONING

Consider an undirected graph $G = (V, E)$, with vertex set V and edge set E, such that $n = |V|$ is the number of vertices and $m = |E|$ is the number of edges. Consider also $c(u)$ and $w(\{u, v\})$ to be the weights of a vertex and an edge respectively.

A solution for the GP problem is a partition of the set of vertices of the graph. This means that we have to divide the vertex set into k disjoint subsets such that their union is equal to the total set. Two partitions are equivalent if for every subset in one partition there exists a subset in the other that is composed of the same vertices. If two partitions are not equivalent, they are different solutions.

To define the problem in a more formal way, let $\mathcal{P} = \{P_1, ..., P_k\}$ be a partition of the vertex set V for a fixed integer k. We can define the set of cut edges as

$$C = \{\{u, v\} \in E | u \in P_i, v \in P_j, 1 \leq i < j \leq k\}, \quad (1)$$

and the cut size as

$$\Theta(\mathcal{P}) = \sum_{\{u,v\} \in C} w(\{u, v\}). \quad (2)$$

Note that the cut size is equal to the number of cut edges if the edge weights are equal to one but different otherwise. Consider also the balance $1 \leq \epsilon \leq k$. Then, the balanced k-GP problem consists on minimizing $\Theta(\mathcal{P})$ subject to the constraints

$$0 < \sum_{u \in P_i} c(u) \leq \frac{\epsilon}{k} \sum_{v \in V} c(v), \quad (3)$$

$$\bigcup_{1 \leq i \leq k} P_i = V, \quad (4)$$

and

$$P_i \cap P_j = \emptyset, \ \forall i, j \in [1, k], \ i \neq j. \quad (5)$$

The first condition ensures that P_i does not exceed the size limit imposed by the balance ϵ and the other two conditions ensure that $\mathcal{P}$ is a partition of V.

III. ADIABATIC QUANTUM COMPUTATION AND QUANTUM ANNEALING

As mentioned in [2], there has been recently much interest in solving NP-complete and NP-hard problems, such as the GP problem, making use of adiabatic quantum computation (AQC) algorithms [4, 5]. They are based on the following idea: consider an initial Hamiltonian H_I, whose ground state is easy to find, and a problem Hamiltonian H_P, whose ground state encodes the solution to a problem of interest. Then, prepare the system to be in the ground state of H_I and adiabatically change the Hamiltonian to H_P. For instance, one could follow the evolution

$$H(t) = A(t/t_f)H_I + B(t/t_f)H_P, \tag{6}$$

for a total time t_f, where $A, B : [0,1] \to \mathbb{R}$ are annealing functions such that $A(0) \gg B(0)$ and $A(1) \ll B(1)$. The evolution of the wave function of the system $|\psi(t)\rangle$ is governed by the time-dependent Schrödinger equation

$$i\hbar \frac{\partial |\psi(t)\rangle}{\partial t} = H(t)\,|\psi(t)\rangle\,. \tag{7}$$

If t_f is large enough, and H_I and H_P do not commute, the system will remain with high probability in its ground state throughout the evolution as long as the conditions of the adiabatic theorem [6] are satisfied. Therefore, measuring the quantum state at time t_f will return the solution to the problem.

Following [5, 7, 8], quantum annealing (QA) pursues the same idea, but restricts H_P to represent a classical objective function. QA algorithms may be physically implemented in open systems susceptible to noise. This may interfere with the computation, increasing the probability of not finishing in the desired ground state. Thus, QA provides a heuristic method for solving combinatorial optimization problems.

We restrict ourselves to QA in the transverse Ising model. The problem Hamiltonian for a system of n spins is written as the quantum version of an Ising spin glass, i.e.

$$H_P = \sum_i^n h_i \sigma_i^z + \sum_{i<j} J_{ij} \sigma_i^z \sigma_j^z, \tag{8}$$

where σ_i^z is a Pauli matrix acting on the ith spin and h_i and J_{ij} are real numbers. We call Eq. (8) an Ising Hamiltonian. Then, we choose the non-commuting initial Hamiltonian to consist of a transverse field

$$H_I = -\sum_i^n \sigma_i^x, \tag{9}$$

where σ_i^x is again a Pauli matrix acting on the ith spin.

The ground state of H_I is an equal probability superposition of all possible states in the eigenbasis of H_P. Note that in such basis the Hamiltonian H_P is diagonal and, thus, for a small number of spins, its ground state can be computed by simple means. However, the dimension of the Hilbert space of such system is 2^n for n spins. Therefore, for a system of n greater than about 50, the problem of finding the ground state of H_P becomes intractable for classical computers. This is when QA has proved to become useful.

The quadratic unconstrained binary optimization (QUBO) model is an alternative formulation to the Ising Hamiltonian. We present it as it will be used further on to implement the GP problem. In this model, the problem Hamiltonian for a system of n particles is defined using an $n \times n$ upper-triangular matrix Q and a vector $\mathbf{x}$ of n binary variables

$$H_P(x) = \sum_i^n Q_{ii} x_i + \sum_{i<j} Q_{ij} x_i x_j, \quad x_i \in \{0,1\} \tag{10}$$

or, more concisely, it can be represented as

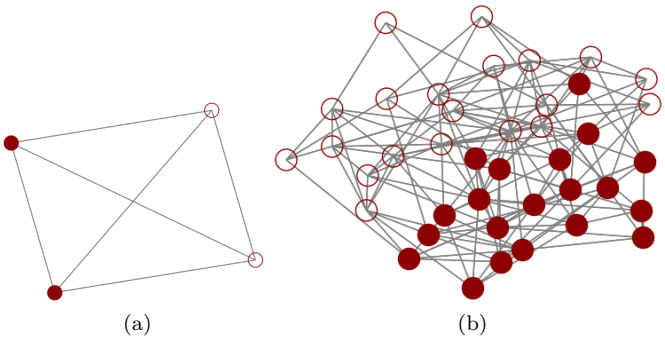
$$H_P(x) = \mathbf{x}^T Q \mathbf{x}. \tag{11}$$

The Ising and the QUBO formulations are related through the identity $\sigma \mapsto 2x - 1$.

A. QUBO formulation of the GP problem

We focus on finding the solution of the balanced GP problem for unweighted graphs, i.e. $c(u) = 1$ and $w(\{u,v\}) = 1$, with $k = 2$ and $\epsilon = 1$. We impose also the additional condition that the number of vertices must be even. In this way, finding a solution to the GP problem consists in dividing the vertex set of a graph into two disjoint subsets such that the number of edges between them is minimized.

In order to be able to write the problem in the QUBO formulation, consider an undirected graph $G = (V, E)$ with n vertices and m edges. The ground state of H_P



(a) (b)

FIG. 1: (a) Unweighted complete graph with 4 vertices. (b) Unweighted graph with 40 vertices and 147 edges. For both (a) and (b) a solution of the GP problem for $k = 2$ and $\epsilon = 1$ is indicated, with filled vertices belonging to one set of the partition and empty vertices to the other set.

should encode the solution to the problem. Moreover, it should minimize the number of cut edges. This can be done as explained in [9]: label each vertex i with $x_i \in \{0,1\}$ depending on the set of the partition they belong to. Then, the number of cut edges is given by

$$N_C = \sum_{(i,j)\in E} (x_i + x_j - 2x_i x_j), \tag{12}$$

subject to the balancing constraint $\sum_{i\in V} x_i = n/2$. The constraint needs to be removed, leading to a relaxation of the problem as follows

$$\min_{\mathbf{x}} \left[\beta \sum_{(i,j)\in E} (x_i + x_j - 2x_i x_j) + \alpha \left(\sum_{i\in V} x_i - \frac{n}{2} \right)^2 \right], \tag{13}$$

where $x_i \in \{0,1\}$, $i = 1, ..., n$. Here, α and β are weight parameters chosen according to [2]:

$$\frac{\alpha}{\beta} \geq \frac{\min(2\Delta, n)}{8}, \tag{14}$$

where Δ is the maximal degree of G. The second term in Eq. (13) can be simplified by removing the constant terms and replacing any squared term with a linear term. Doing so, the problem Hamiltonian in the QUBO formulation is given by

$$\begin{aligned} H_P = \ & \beta \sum_{(i,j)\in E} (x_i + x_j - 2x_i x_j) \\ & + \alpha \left(\sum_{i\in V} (1-n) x_i + \sum_{i\in V} \sum_{i<j} 2x_i x_j \right), \end{aligned} \tag{15}$$

where $x_i \in \{0,1\}$, $i = 1, ..., n$. This can be more concisely expressed with the matrix

$$Q_{ij} = \begin{cases} 2(\alpha - \beta), & (i,j) \in E \\ 2\alpha, & (i,j) \notin E,\ i \neq j \\ \beta\delta_i - \alpha(n-1), & i = j \end{cases} \tag{16}$$

where δ_i is the degree of the vertex i.

B. Numerical simulation

To exemplify how the formulation of section III A can be used to solve a problem, we now present simulation results for the complete 4-vertex graph of Fig. 1a, with $\alpha = 1.25$ and $\beta = 1$. The code used in the simulations is available at [10]. Following Eq. (6), the simulation of the evolution is performed by numerically solving Eq. (7) using the Crank-Nicolson algorithm. Therefore, we set

$$|\psi(t+\Delta t)\rangle = \left(1 + i\frac{\Delta t}{2\hbar} H(t)\right)^{-1} \left(1 - i\frac{\Delta t}{2\hbar} H(t)\right) |\psi(t)\rangle \tag{17}$$

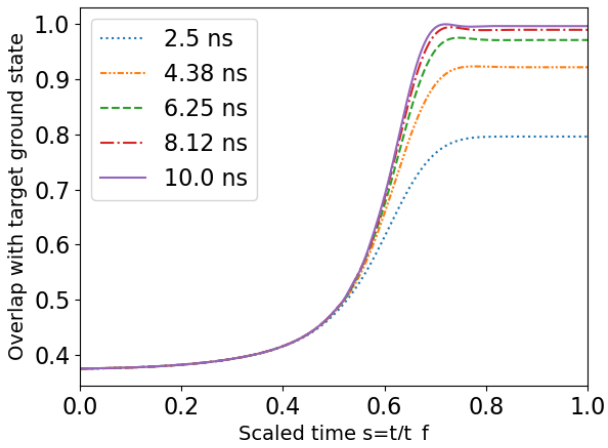


FIG. 2: Overlap of the instantaneous ground state, parametrised by $s = t/t_f$, with the target states for the graph of Fig. 1a as explained in section III B. Values of t_f range from 2.5 ns to 10 ns. Annealing functions $A(s)$ and $B(s)$ implemented in the DW_2000Q_6 annealer are used in the evolution.

and iterate.

We are able to study the evolution of the overlap with the target ground states for different values of annealing time t_f ranging from 2.5 ns to 10 ns. The overlap is defined as $\sum_{|\varphi\rangle} |\langle\psi(s)|\varphi\rangle|^2$, where $|\psi(s)\rangle$ is the instantaneous ground state parametrized by $s = t/t_f$, and $|\varphi\rangle$ ranges over all six ground states of H_P. Results are presented in Fig. 2. It can be seen that larger t_f values give better overlap results at the end of the evolution. Actual annealing functions $A(s)$ and $B(s)$ implemented in the DW_2000Q_6 annealer are used to perform the evolution.

IV. EXPERIMENTAL QUANTUM ANNEALING

In this section we implement the GP problem on D-Wave's DW_2000Q_6 annealer [3]. We use the example of section III B to understand how problems are implemented in the system. Then, we explore GP for larger problems.

It is useful to understand D-Wave's quantum hardware architecture as an undirected graph $U = (V_U, E_U)$, often called working graph, with weighted vertices and weighted edges. Each vertex $i \in V_U$ corresponds to a physical qubit and each edge $(i,j) \in E_U$ corresponds to a coupler between qubits i and j. We can distinguish between two available annealers, for which we are offered one minute of computational time: the DW_2000Q_6 annealer, with the Chimera graph ($\sim$ 2000 qubits); and the Advantage systems, with the Pegasus graph ($\sim$ 5000 qubits).

A. Minor-embedding

One can also understand Eq. (8) as an undirected graph $G = (V_G, E_G)$. Then, each spin corresponds to a vertex (or a logical qubit) with weight h_i and each edge has a weight J_{ij} associated. Similarly, for Eq. (10) weights are stored as coefficients in Q. A problem can only be solved using D-Wave's annealers if G can be embedded as a subgraph of the working graph U. This is referred to as the minor-embedding problem [11, 12].

The basic idea is to find a subgraph G_{em} of U such that G can be obtained from G_{em} by contracting edges. More precisely, let $U = (V_U, E_U)$ be a fixed graph. Given a graph $G = (V_G, E_G)$, the minor-embedding of G in U is a map $\phi : V_G \to \mathcal{P}(V_U)$ such that:

- $\phi(u)$ is a connected subgraph for $u \in V_G$;
- $\phi(u)$ and $\phi(v)$ are disjoint for $u \neq v$; and
- if $(u, v) \in E_G$ there is at least one edge between $\phi(u)$ and $\phi(v)$ in U.

If ϕ exists we say that G is embeddable in U and G is called a minor of U.

Logical qubits in G can be modeled by any collection of physical qubits in U provided they form a connected subgraph called a chain. In a chain, physical qubits are coupled such that they function as a single qubit. This can be achieved by setting the coupler weights between them to a reasonably large negative value, whose absolute value is known as `chain_strength`. This value should be chosen carefully: if it is too strong it may slow down the adiabatic process but if it is not large enough it may not make the linked qubits act as a single one. Then, logical weights are distributed among the remaining available physical components.

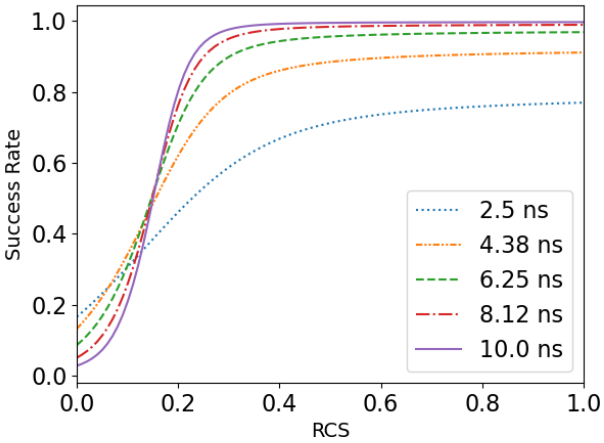


FIG. 3: Success rate vs RCS for the embedding of the graph of Fig. 1a in the Chimera graph as explained in section IV A. Values of t_f range from 2.5 ns to 10 ns.

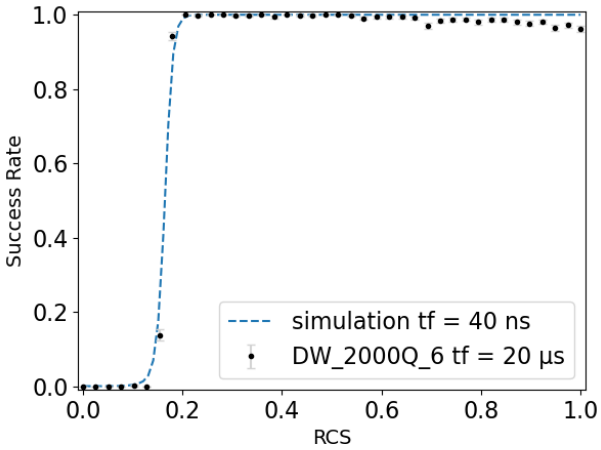


FIG. 4: Success rate vs RCS on DW_2000Q_6 annealer for the embedding of the graph of Fig. 1a in the Chimera graph as explained in section IV A. Simulation results with $t_f = 40$ ns are fitted with actual annealing results with $t_f = 20$ μs and 500 repetitions.

Following [13], we can define `chain_strength` in terms of a relative chain strength RCS $\in [0, 1]$ according to

$$\texttt{chain\_strength} = \text{RCS} \times \texttt{max\_strength}, \tag{18}$$

where $\texttt{max\_strength} = \max(\{|h_i|\} \cup \{|J_{ij}|\})$ for the Ising model and $\texttt{max\_strength} = |\max(Q)|$ for the QUBO model. When defining these variables, one should take into account that D-Wave's annealers do not accept arbitrary values of h_i, J_{ij} and Q (see [3]).

To study which value of RCS should be used, we embed the graph of Fig. 1a in the Chimera graph for different values of RCS and perform the evolution of Eq. (6). Results are given in Fig. 3, where the success rate corresponds to the value of the overlap, as defined in section III B, at the end of the evolution [10]. We observe that an optimal RCS value should be greater than 0.2.

Actual annealing results on the DW_2000Q_6 annealer with annealing time $t_f = 20$ μs and 500 repetitions are presented in Fig. 4. These are shown together with simulation results for $t_f = 40$ ns, $\alpha = 1.25$ and $\beta = 1$ [10]. Despite the huge difference in time, the simulation fits the points remarkably well. This phenomenon can be explained, following [14], by taking into account that the simulations are of an ideal closed system, while the quantum annealer is open and susceptible to noise. However, one thing that still holds is the behavior of the success rate vs the RCS value as simulated in Fig. 3.

B. Quantum and classical approach

As a final application, we explore the GP problem as defined in section III A on the DW_2000Q_6 annealer for graphs with sizes ranging from 20 to 60 vertices [10]. All

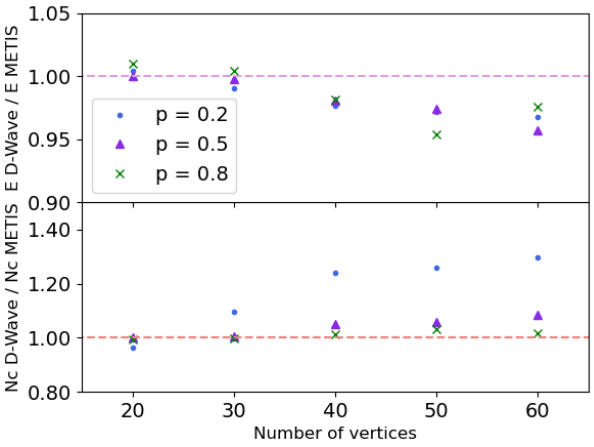


FIG. 5: Comparison of the solutions found by the DW_2000Q_6 annealer and METIS for graphs of different sizes and connectivity. The number of cut edges (N_C) and the energies of the solutions (E) obtained by both methods are studied.

graphs are randomly generated according to Erdös-Rényi model [15] with probabilities $p \in \{0.2, 0.5, 0.8\}$, corresponding to sparse, balanced and dense graphs, respectively. An example of a graph with $n = 40$ and $p = 0.2$ is shown in Fig. 1b. We fix RCS $= 0.25$ and $t_f = 200$ μs and repeat the annealing 500 times.

The quality of GP solutions is evaluated by comparison to METIS, a multilevel GP classical framework [16]. We compute the number of cut edges for each partition and the energy of the solutions, corresponding to the energy of the ground state of H_P. For every (n, p) value, we study two different graphs and plot the average results in Fig. 5. We are able to obtain comparable results in the number of cut edges for all graphs of 20 and 30 vertices. For dense graphs ($p = 0.8$), results are of similar quality for all studied graphs as the ratios are close to one. The tendency of the energy plot confirms that, for balanced partitions, less number of cut edges implies more negative ground state energy values.

V. SUMMARY AND CONCLUSIONS

In this work, we solved the graph partitioning problem by making use of quantum annealing. We were able first to obtain simulation results of the DW_2000Q_6 annealer for the 4-vertex graph of Fig. 1a with high overlaps (> 0.95) for annealing times $t_f \gtrsim 6.25$ ns. To physically implement the problem, we computed a minor-embedding in the Chimera graph and simulated the embedded problem. We were able to fit simulation results with $t_f = 40$ ns with results from an actual quantum annealer with $t_f = 20$ μs and 500 repetitions. We could also simulate how the success rate varies with the RCS value, obtaining maximum success rates for RCS ≥ 0.2. Finally, we studied the quality of the solutions obtained by the DW_2000Q_6 annealer by comparison to the solutions given by METIS. The performance of the quantum annealer for dense graphs was of particularly good quality as compared to existing classical algorithms. In a future work, having more computational time, we would extend the analysis to larger graphs.

Acknowledgments

I would like to thank Dr. Bruno Juliá Díaz and Abel Rojo Francàs for their orientation and advice throughout the elaboration of this work. Also a special mention to friends and family for their support, which came in many ways.

[1] A. Buluc, H. Meyerhenke, I. Safro, P. Sanders and C. Schulz, Algorithm engineering, 117–158 (2016).
[2] A. Lucas, Frontiers in Physics 2, 5 (2014).
[3] D-Wave System Documentation: `https://docs.dwavesys.com`
[4] R. LaPierre, *Introduction to Quantum Computing*, (Springer, 2021).
[5] C. C McGeoch, *Adiabatic Quantum Computation and Quantum Annealing: Theory and Practice*, (Morgan & Claypool Publishers, 2014).
[6] T. Albash and D. A. Lidar, Rev. Mod. Phys. 90, 015002 (2018).
[7] E. Cohen and B. Tamir, Eur. Phys. J. Special Topics, 224 (1), 89–110 (2015).
[8] T. Kadowaki and H. Nishimori, Phys. Rev. E 58, 5355 (1998).
[9] H. Ushijima-Mwesigwa, C. F. Negre and S. M Mniszewski, *Graph Partitioning Using Quantum Annealing on the D-Wave System*, Proceedings of the Second International Workshop on Post Moores Era Supercomputing, 22–29 (2017).
[10] Graph-Partitioning-QA: `https://github.com/AndreaPeMa0/Graph-Partitioning-QA.git`
[11] V. Choi, Quantum Inf. Process. 7 (5), 193–209 (2008).
[12] V. Choi, Quantum Inf. Process. 10 (3), 343–353 (2011).
[13] D. Willsch, M. Willsch, C. D. Gonzalez Calaza, F. Jin, H. De Raedt, M. Svensson and K. Michielsen, Quantum Inf. Process. 21, 141 (2022).
[14] M. Willsch, D. Willsch, F. Jin, H. De Raedt and K. Michielsen, Phys. Rev. A 101, 012327 (2020).
[15] B. Bollobás, *Random Graphs*, (Cambridge University Press, 2001).
[16] G. Karypis and V. Kumar, SIAM Journal on Scientific Computing 20 (1), 359–392 (1998).