Evaluation of the optimal parameters of quantum annealing for the solution of matching problems

Daniel Vert

Deep Tech Ecosystem, SYSTEMATIC, Palaiseau, France
daniel.vert@systematic-paris-region.org

1 Introduction

Inspired by the simulated annealing algorithm, quantum annealing is a heuristic using quantum fluctuations that allows (theoretically) to cross large energy barriers by tunneling. This particularity gives it a clear advantage over its classical counterpart. In recent years, two new generations of quantum annealing from the company D-Wave are available : Advantage with more than 5000 qubits and 15 couplings per qubit and Advantage2.0 with the topology named « Zephyr »of 500 qubits but with 20 connections between the qubits. One of the main limitations that « greatly »affect performance is the connectivity between qubits. Due to too little connectivity, the variables of a problem are usually not directly integrable on the architectures. One must then resort to an integration of the instance graph into the hardware graph by doing a "qubit duplication". Determining a minimal subgraph of the machine's toplogic graph, and isomorphic to our instance, is generally a difficult problem in itself. Nevertheless, there are heuristics to find embeddings that limit excessive duplication of qubits.

The fact of having duplication implies post-processing. Moreover, to avoid duplication errors (two duplicated qubits that do not have the same value in output), we can impose a parameter (`chain_strength`) corresponding to the weight of the matching between duplicated qubits.

In this context, we investigate to what extent the embbedding that minimizes the number of duplicated qubits and the `chain_strength` influence the results on the matching problem [1]. To do so, we try to obtain the graph with the least number of duplications and try to constrain the qubit chains as much as possible to limit the duplication errors on the Advantage machine.

2 Study of the parameters

2.1 Embedding

Generating the optimal embedding of one graph on another is a difficult problem in itself. To solve this problem a probabilistic algorithm is used. This means that each time we call the function, it can return a different embedding with a different number of duplications. In our case, it is interesting to observe the minimal embedding (with the least number of duplications) to observe the influence of the number of qubits used on the quality of the solution and thus generate several different embeddings. In the previous study, we could see that the more physical qubits we used, the more difficult it was for D-Wave to solve the problem since we obtained a high probability of having duplication errors. Indeed, the physical qubits of a chain (resulting from the duplication of qubits) must behave as a single entity (a logical qubit) and thus must all have the same value at the end of the annealing process.

2.2 `Chain_strength`

To achieve this, the J_{ij} couplings between the physical qubits are set to a large negative amplitude value. The amplitude determines the strength of the matching of these qubits and the ease with which the chain can « break ». In this context, a « chain break »means that different qubits of the same chain end up in different states even though they represent the same logical qubit. Too many chain breaks should be avoided because the returned solutions may become random and could be of poor quality [2]. The optimal value of the chain strength depends on the problem and possibly on the embedding. If the chain strength is chosen too weak, it is energetically favorable to break it. On the other hand, if it is chosen too strong, all the parameters defining the problem instance will be very small compared to the `chain_strength` and therefore they will be negligible for the algorithm.

3 Implementation on D-Wave and analysis of results

In order to answer this question, we chose the bipartite matching problem of maximum cardinality $\mathbf{G_n}$ from the paper by Sasaki and Hajek [1]. They studied the family of special cases of the problem which is polynomial, but with a mathematical expectation of the number of iterations required by a large class of (classical) annealing type algorithms to reach maximum matching in $O(e^n)$. With this problem we can verify that the D-Wave obtains the optimal solution and our protocol is the following : we first leave the default parameters of the D-Wave in order to solve the problem on all sizes of G_n that it is possible to integrate. The initial parameters are then : an annealing time of 20μs, a `chain_strength` of 1 and 100 randomly drawn embeddings. By taking the embedding which obtains the best results, we can see that from 64 variables, it is necessary to use more than 170 qubits to obtain only 2 times the optimal solution on the 100 runs launched. Beyond G_4, the gap between the optimal solution and the lowest cost solution found becomes too important (between 15% and 40%) to be interesting. With this first observation, we can observe that D-Wave with more than 5000 qubits can tackle problems with a maximum size of 512 variables but does not obtain good results beyond a certain instance size.

Now, we reconfigure the parameters in order to improve our first results, we adjust the `chain_strength` as well as possible, we set the annealing time to 2000μs (maximum time imposed by the machine) and we determine the embedding with the least duplication possible for each instance size. We can see that the results obtained are much better than before. For G_2 we are at more than 90% of success against barely 30% and for G_3, we pass from 2% to 21% with less than 140 qubits (against 174 before).

Our results suggest that determining embedding with as few duplications as possible and a large annealing time allow us to obtain much better quality results in all instance sizes of at least 20%.

Références

[1] Daniel Vert. *Performance evaluation of quantum annealing on bipartite matching instances.* 32nd EURO Conference, Aalto University.

[2] Willsch, Madita and Willsch, Dennis and Michielsen, Kristel. *Lecture Notes : Programming Quantum Computers.* Quantum Physics, Computational Physics, ArXiv