Quantum Computing applied to optimization problems

Pablo Diez-Valle

In collaboration with Juan José García Ripoll

Consejo Superior de Investigaciones Científicas - CSIC, Spain

pablo.diez@csic.es — (+34) 91 568 02 44

Motivation

It is well known the usefulness of optimization on a wide range of disciplines. There is a broad spectrum of combinatorial optimization problems which can be represented as a Quadratic Unconstrained Binary Optimization problem (QUBO). QUBO problems consist in the maximization or minimization of a cost function F defined as follows:

$$F(x) = \sum_{i,j} x_i Q_{i,j} x_j \;\; ; \quad x \in \{0,1\}^n \,, \tag{1}$$

where x is a n-vector of binary variables, and Q is a n-by-n square symmetric matrix of coefficients.

The utility of this model has been extensively discussed in the literature. For instances, representing and solving optimization problems on graphs, resources allocation problems, clustering problems, satisfiability, sequencing and ordering problems, various forms of assignment problems etc. Some applications appear naturally as QUBO, however other optimization problems can be transformed conveniently into the expression (1). In our research we shall focus on the resolution of QUBO problems using quantum techniques.

From QUBO to the Ising model

The QUBO problem can always be mapped to the Ising model. Thereby the optimization of the cost function F (1) becomes a search of the minimum energy state of an Ising Hamiltonian. Doing the following change of variable $x_i \to \frac{1}{2}(1+\sigma_i^z)$, where σ_i^z is the Pauli matrix, we can express the previous function as:

$$\hat{f}(\sigma_i^z) = -\frac{1}{2}\hat{H} + cte, \tag{2}$$

where $cte = \frac{1}{4}\sum_{i,j} Q_{ij}$, and $\hat{H}$ correspond to an Ising Hamiltonian:

$$\hat{H} = \sum_{i<j} J'_{ij}\sigma_i^z\sigma_j^z + \sum_i J''_i\sigma_i^z, \tag{3}$$

with $J'_{ij} = -Q_{ij}$ and $J''_i = -\sum_j Q_{ij}$. Now $x_i = 0, 1$ correspond to the states of σ_i^z with eigenvalue -1, +1 respectively. Thus, we need to find the ground state of the Ising Hamiltonian (3) in order to maximize our cost function F.

Objective / Hybrid quantum-classical algorithm

Our goal is to design and to evaluate hybrid quantum-classical algorithm to find an optimal (or good enough) solution to the QUBO problem.

The emerging paradigm for the resolution of optimization problems using quantum techniques is the known as hybrid quantum-classical algorithm [2]. This kind of algorithms combines quantum computing via the variational method, and classical computing to optimize the variational parameters. As is shown in Figure 1, in the first place the quantum device produces a ansatz (which depend on the specific algorithm implemented) that depends on the θ parameters,

$$|\Psi(\theta_i)\rangle = U_L(\theta_L)...U_1(\theta_1)|\text{initial state}\rangle \tag{4}$$

The ansatz (4) is the trial state prepared through the application of L unitary transformations. Hereafter we shall refer to L as the layers or the depth of the quantum circuit. Then, the ansatz is used to calculate the expectation values of the Hamiltonian (defined by the problem). These results are used by the classical computer to calculate and to optimize the cost function, returning new θ values. This process is iterated until we obtain a good solution.

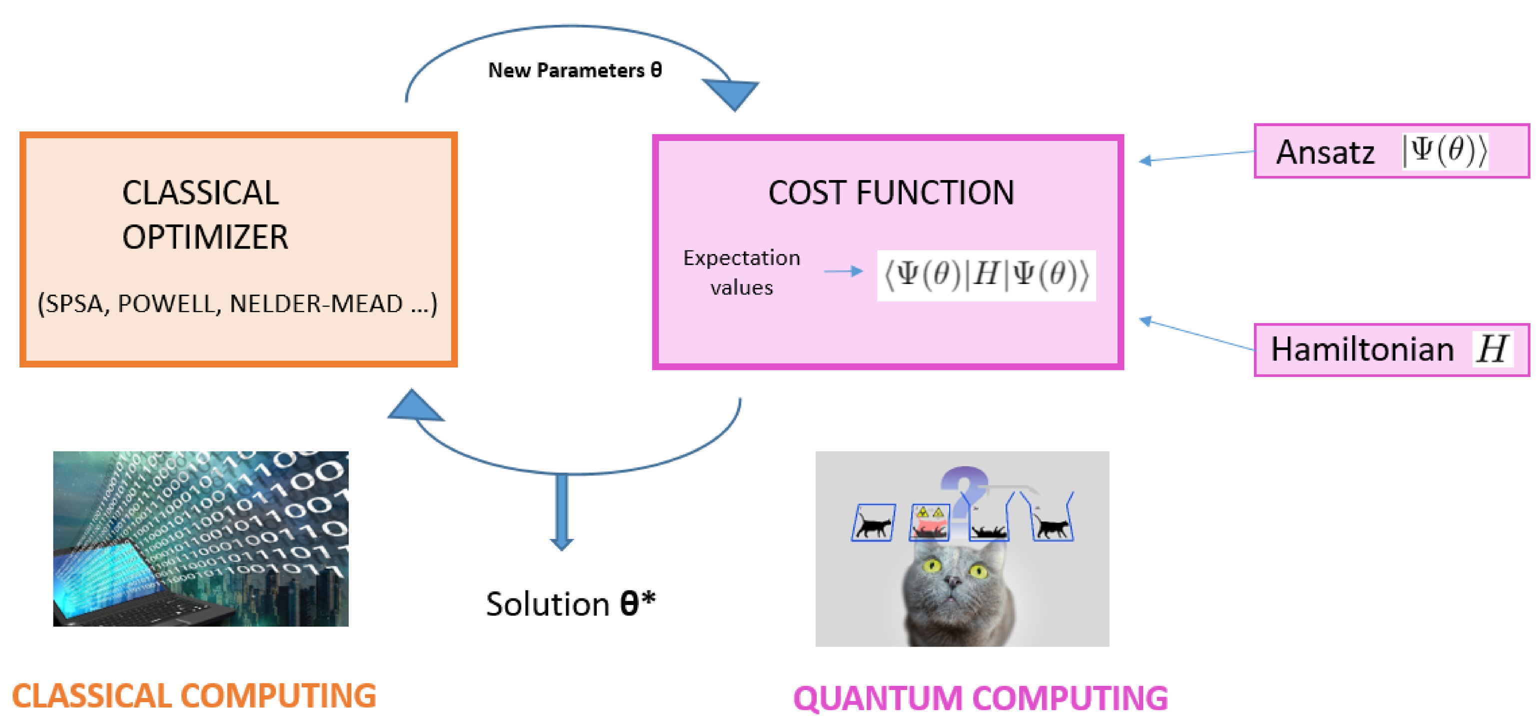

Figure 1: Operating diagram of an hybrid quantum-classical algorithm based on the variational method.

Results

Our first objective is the study of the efficiency of hybrid optimization algorithms, using the speed and the accuracy as indicators. This research, performed in quantum simulators of ideal quantum computers, shall prepare for the Noisy Intermediate-Scale Quantum (NISQ) technology which will be available in the near future.

When we program an hybrid algorithm for the minimization of a given Ising Hamiltonian (3), we have three variables which we can control: the depth of the circuit (L), the ansatz ($|\Psi(\theta)\rangle$, controlled by the unitary operators (U_i) we apply on the initial state (4)) and the classical optimization method (Powell, Cobyla, Nelder-Mead...). We show our work in progress in which we study different algorithms modifying the mentioned conditions.

Qiskit

Qiskit is an open-source software development kit provided by IBM which enables to create quantum computing programs in Python Programming Language [1].

Qiskit includes a module (*Qiskit Aqua*) which contains algorithms for near-term quantum applications, for instance optimization applications. In the first place, we test the hybrid quantum-classical algorithms that are provided by Qiskit, namely, Variational Quantum Eigensolver (VQE) and Quantum Approximate Optimization Algorithm (QAOA).

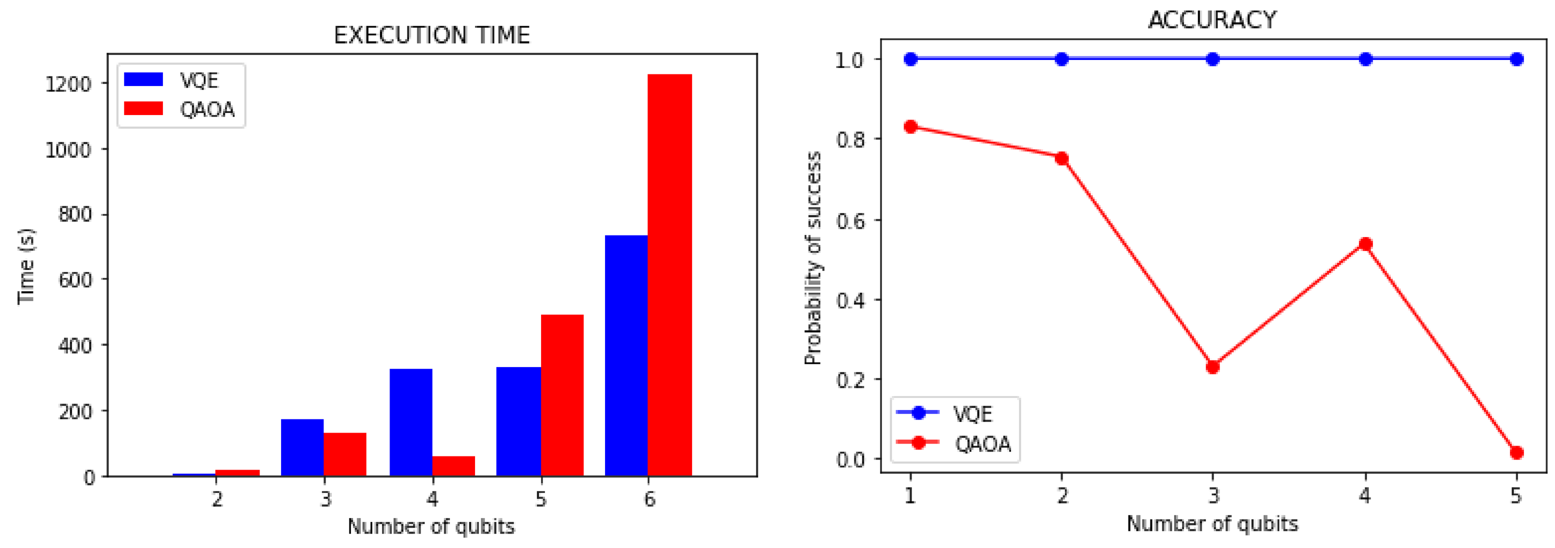

Figure 2: Results obtained running the hybrid quantum-classical algorithm provided by Qiskit, namely, VQE and QAOA. We test their efficiency modifying the number of qubits. The classical optimizer used is Constrained Optimization BY Linear Approximation (COBYLA).

Homemade simulator

As we can see in figure (2) the execution time has an exponential increase with the number of qubits, while the results get worse. In order to extrapolate our algorithms to a large number of qubits, we would need to reduce the evaluation time of the algorithm. However, the use of Qiskit has a limitation for our research: we have a lot of overhead in the code that we do not control and which affects the efficiency of our computation. For this reason, we create our own hybrid quantum-classical optimizer.

As a first approach, we need to test our homemade simulator. We run it using the known ansatz of VQE, and we compare the results with those obtained with Qiskit.

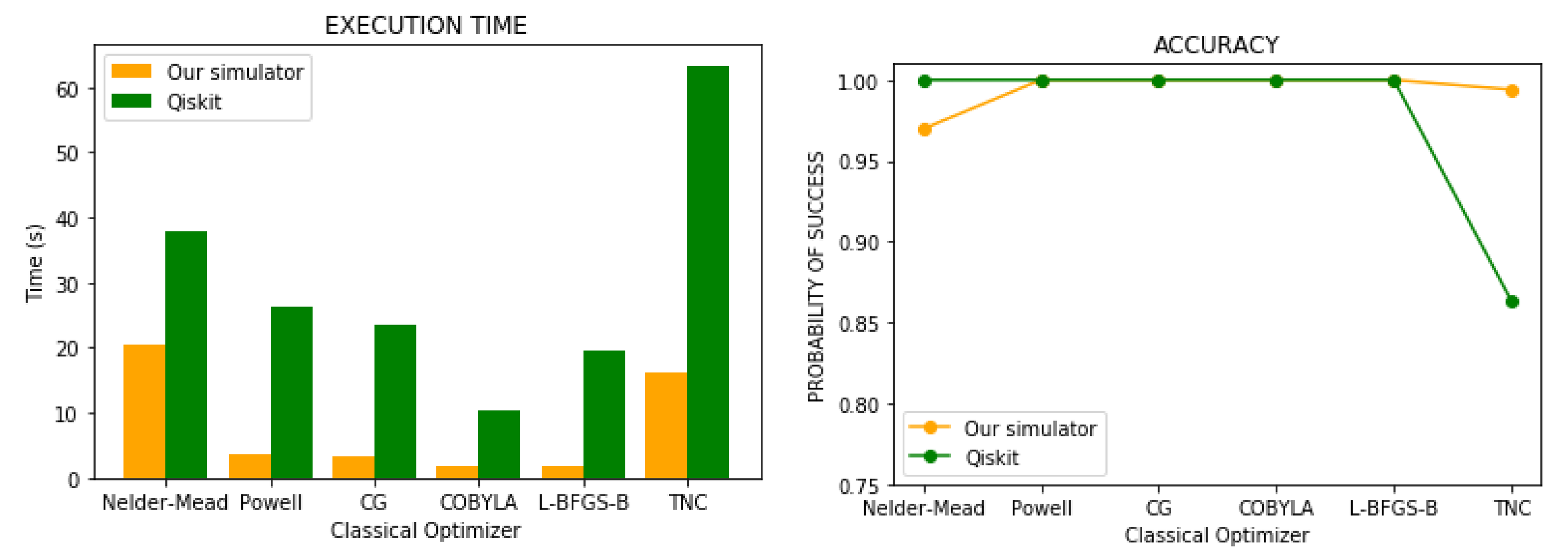

Figure 3: Comparison between the results obtained running VQE algorithm provided by Qiskit, and the algorithm of our homemade simulator. We modify the classical optimization method. The circuit was created with five qubits and two layers.

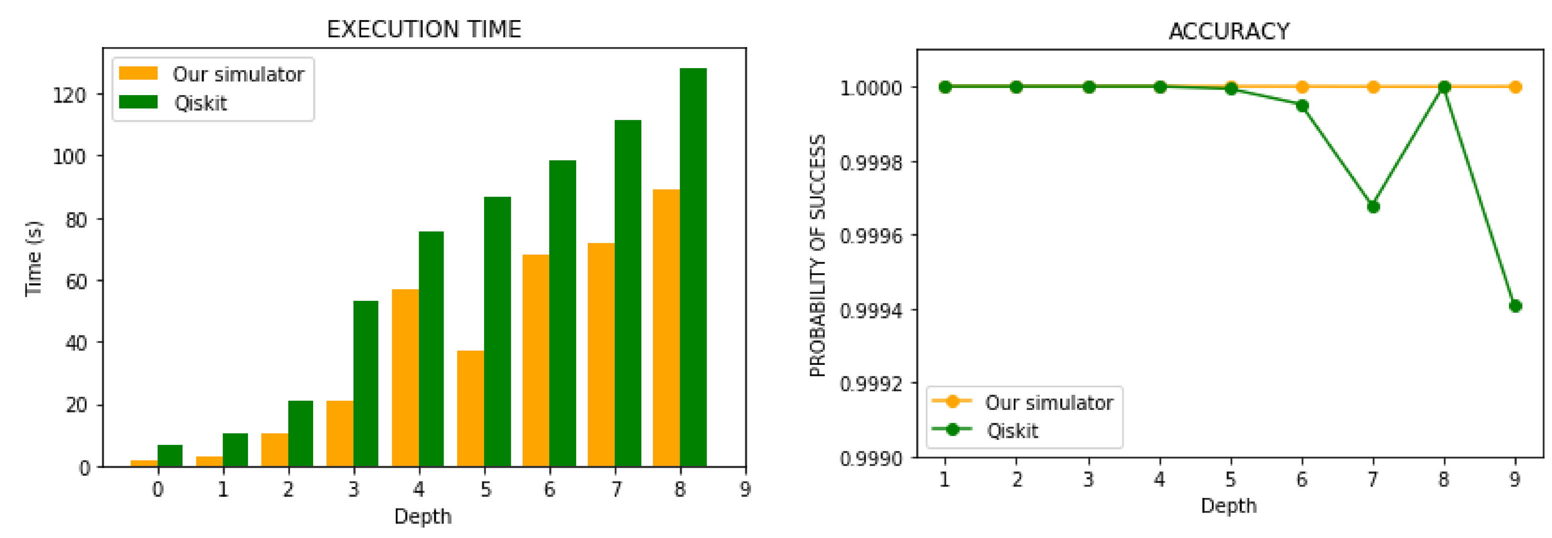

Figure 4: Comparison between the results obtained running VQE algorithm provided by Qiskit, and the algorithm of our homemade simulator. We modify the depth of the variational form. The circuit was created with five qubits and the classical optimizer is Limited-memory Broyden-Fletcher-Goldfarb-Shanno Bound (L-BFGS-B).

Current research

In the light of the test results we are using our homemade optimizer to study different hybrid algorithms. Specifically, we try a variety of trial states. For instance,

- $U(\theta) = \prod_l^{layer}\left[\prod_N^{\text{number qubits}} e^{i\theta_{Nl}\sigma_N^y} \cdot U_{\text{entangler}}\right]$. Where $U_{\text{entangler}}$ is a collection of fully entangling gates. This choice corresponds to the variational form known as RY, and it is implemented in Qiskit.
- $U(\theta,\gamma) = \prod_l^{layer}\left[\prod_N^{\text{number qubits}} e^{i\theta_{Nl}\sigma_N^y} \cdot e^{i\gamma_l H}\right]$. Where H is the Hamiltonian of the problem, and θ_{Nl}, γ_l variational parameters. Note that in this algorithm, as well as in QAOA, the Hamiltonian operator play a dual role. In the one hand it is the object which need to be optimized. In the other hand, the Hamiltonian is part of the unitaries that govern the evolution of the system.

Summary

We implemented our own homemade quantum simulator in order to design more efficient optimization algorithms based on the variational method. Future work will be aimed to increase speed and accuracy of hybrid quantum-classical optimization algorithms, and to explore the effect of entanglement on it.

References

[1] https://qiskit.org.

[2] Nikolaj Moll et al. *Quantum Sci. Technol*, 3, 2018.

Acknowledgements: I thank INFIERI Summer School and Huazhong University of Science and Technology (HUST) for support my attendance at 5th INFIERI in Wuhan (China).