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:

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

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

The utility of this model has been extensively discussed in the literature. For instance, 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 $F$ becomes a search of the minimum energy state of an Ising Hamiltonian. Doing the following change of variable

$$x_i = 2x_i - 1,$$

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.

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 hybrid method. Here- after 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 evaluated by the classical computer to calculate and to optimize the cost function, returning new $\theta$ values. This process is iterated until we obtain a good solution.

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).

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, $U(\theta) = \prod_{l=0}^{L} \prod_{i=\text{number qubits}} \exp(\theta_i \text{cnot}_{ij} + \text{entangler}_{ij})$. Where $\text{cnot}_{ij}$ is in a collection of fully entan-
ging gates. This choice corresponds to the variational form known as $\text{RY}$, and it is implemented in Qiskit.

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.

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

References

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