

Hybrid Quantum Computing

Spencer Obiero*

April 11, 2024

Abstract

The idea of harnessing the power of atoms for computation has been around for so long. However, it wasn't until in 1994 when Peter Shor (1997) invented the factoring algorithm and in 1995 when Lov Grover (1996) invented the Grover Search Algorithm that people's interest for realizing quantum computers got bigger. Many people became intrigued by the futuristic vision of developing quantum computers. Moreover, the thought of breaking the RSA code was becoming a reality¹; people thought that quantum computers would be able to enhance this quest and many more. Consequently, more research kicked off and since then, there have been many quantum algorithms developed speculating the utilization of quantum computing. The problem, however, is that the realization of actual quantum computers has proven to be really hard due to noise and other physical limitations, one of which is described in this paper. While there are firms such as Google and IBM, to mention a few with actual quantum computers, these machines face the problem of decoherence. Decoherence is the tendency of a quantum state to be short lived due to environmental interactions. Because of short coherence, many algorithms are deemed to be futile as not much can be done in a short period of time. For instance, simulating a certain material might involve a lot of time and memory as demonstrated by Aspuru-Guzik et al. (2005), and this might be hard to do with the time constraint posed by decoherence in quantum systems.

This paper demonstrates ideas that involve combining both classical and quantum computing to circumvent the challenge of decoherence. We particularly look at a simple demonstration of quantum simulation, where we show how to prepare a specific state using quantum circuits. We then use classical algorithms such as optimization to get the best values to parameterize the state we would like to use in our simulation. This process is iterated until global properties of our desired simulations are achieved.

1 Introduction

Prior to Shor and Grover, the fathers of quantum simulations such as R. P. Feynman (1982) and Seth Lloyd (1996) envisioned the idea of using variational circuits to utilize the power of quantum states. There are many ways to approach this problem.

The first one involves superconducting architecture, in which microwave pulses are used to control the speeds of items used in computation. At low temperatures, it is easy to harness the power of quantum computation because of the behavior of the atom but this is costly due to the cooling requirement of the system. Moreover, it is hard to achieve a large quantum circuit depth due to to

complexity that comes with quantum gates such as the H and the Controlled-NOT as demonstrated by Isenhower et al. (2010).

The second involves trapped ions, in which charged atomic particles are confined in a magnetic field, and the spins of the atoms used as our basis for computation. This mechanism has promising speculations but as seen in the papers by Hempel et al. (2018), simulating simple gates such as CNOT, though achievable, involves a lot of complexity making it hard to scale up.

Photonic polarization is another technique which as seen in this paper by Peruzzo et al. (2013), can be done in room temperature thus overcoming the cost of cooling associated with superconducting architectures. The problem with photons, however, is

*sobiero@bates.edu

¹In his paper, Shor speculates this possibility but then points out the problem of noise and decoherence as limiting factors

that they are easy to escape, and cannot be stored for future computation. Moreover, it is hard to control for background noise when conducting experiments with photons, even if it is in a dark room.

While quantum scientist are still in the process of researching for better ways to improve quantum computation to achieve optimum utility, we can incorporate quantum ideas into classical computation and combine the two to potentially get the best of both worlds. Below, we demonstrate an example of how this could be done by solving an optimization problem, for finding the low cost parameters for a quantum state, given a mixing and cost hamiltonian.

2 Methodology

While describing a quantum system, a hamiltonian is used to specify the energy descriptors for the system, e.g. a hamiltonian involving terms that indicate magnetic properties used to characterize interaction between nearest neighbors of particles within our quantum system may be used for the quantum ising model. For this problem, we use simple mixing and cost hamiltonian represented by a Pauli X and Pauli Z respectively to represent the bias we want to minimize - this is a simplification of the hamiltonian that Verdon et al. (2019) uses in their paper on quantum algorithms for supervised learning. With quantum circuit analysis, it can be shown that our desired hamiltonian can be represented by the circuit below:

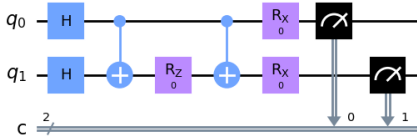


Figure 1: representing our hamiltonian in a simple circuit parameterized by $\beta = \alpha = 0$

We use scipy's minimize() function to optimize the parameters that potentially characterize the state that minimizes the bias hamiltonian. After we do the optimization, we see that the state we want is: $|00\rangle$ or $|11\rangle$ of the two as shown in the figure 2

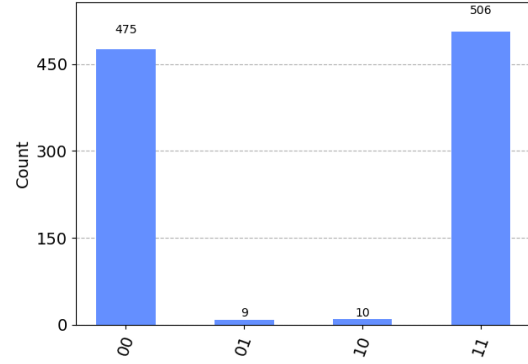


Figure 2: the $|00\rangle$ and $|11\rangle$ kets minimize our cost hamiltonian

3 Results

As expected, the states that minimize our cost hamiltonian are the ones shown above ($|00\rangle$ & $|11\rangle$); these are the eigenvalues of the Pauli Z matrix which in this case represents the bias hamiltonian. This seems trivial but for illustrative purposes it works out fine. For practical purposes, more involved and complex hamiltonians are used. Getting these types of hamiltonians and encode them into mathematical objects requires some expertise in quantum mechanics and advanced quantum computation. Once the hamiltonian is obtained, one proceeds to the preparation of the ansatz state as demonstrated above. Finally, the punchline in our simple algorithm is in the parameterization of the ansatz states with regard to the hamiltonian in question; that's where the classical optimization comes in handy. With more complex hamiltonians, the heuristic idea remains the same - the difference could only manifest when encoding the hamiltonian into a matrix or circuit. This is the idea that builds the foundation of Quantum Variational Eigensovers (VQE) as demonstrated by Hempel et al. (2018) and Peruzzo et al. (2013). Moreover, Alam et al. (2019) uses this idea to implement the quantum approximation optimization algorithm, which is a tool that is being developed for quantum machine learning and is a gem specifically in training neural networks.

4 Discussion

As demonstrated above, our algorithm works for a simple case of using Pauli X and Pauli Z to represent our hamiltonian. Again, with different hamiltonians, the expectation is that the algorithm still works but more thought needs to be put in the encoding of data.

A realization of this simple model may be done using ion traps, which provide well defined hamiltonians that can be used to describe the system. This choice gets rid of the disadvantage associated with cooling resources thus bringing us nearer and nearer to quantum computing/simulation actualization.

With more research into decoherence and physi-

cal realization of quantum systems, the potential of scalability might be attained. A drawback, however, is that some hamiltonians may take a longer time to minimize, thus making things hard when the simulation involves a bigger number of parameters for these complicated hamiltonians, even with the physical realization available to us.

To wrap up, we reflect on what Feynman thought of quantum simulation. As he puts it in his 1982 paper: quantum simulation will give us the ability to explain quantum phenomena, staying consistent with all the known facts and suggesting new facts that are yet to be explained. With these quantum algorithms in place and future discovery of better ways to realize quantum computation, we will eventually realize this idea.

References

- Alam, M., Ash-Saki, A., and Ghosh, S. (2019). Analysis of quantum approximate optimization algorithm under realistic noise in superconducting qubits.
- Aspuru-Guzik, A., Dutoi, A. D., Love, P. J., and Head-Gordon, M. (2005). Simulated quantum computation of molecular energies. *Science*, 309(5741):1704–1707.
- Eleanor Rieffel, Wolfgang Polak (2014). *Quantum Computing: A Gentle Introduction*. MIT Press paperback edition.
- Feynman, R. P. (1982). Simulating Physics with Computers. *International Journal of Theoretical Physics*, 21(6-7):467–488.
- Grover, L. K. (1996). A fast quantum mechanical algorithm for database search.
- Hempel, C., Maier, C., Romero, J., McClean, J., Monz, T., Shen, H., Jurcevic, P., Lanyon, B. P., Love, P., Babbush, R., Aspuru-Guzik, A., Blatt, R., and Roos, C. F. (2018). Quantum chemistry calculations on a trapped-ion quantum simulator. *Physical Review X*, 8(3).
- Isenhower, L., Urban, E., Zhang, X. L., Gill, A. T., Henage, T., Johnson, T. A., Walker, T. G., and Saffman, M. (2010). Demonstration of a neutral atom controlled-not quantum gate. *Phys. Rev. Lett.*, 104:010503.
- Lloyd, S. (1996). Universal quantum simulators. *Science*, 273(5278):1073–1078.
- Michael A. Nielsen, Isaac L. Chuang (2016). *Quantum Computation and Quantum Information*. 10th Anniversary Edition.
- Peruzzo, A., McClean, J. R., Shadbolt, P. J., Yung, M.-H., Zhou, X., Love, P. J., Aspuru-Guzik, A., and O’Brien, J. L. (2013). A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5.
- Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509.
- Verdon, G., Broughton, M., and Biamonte, J. (2019). A quantum algorithm to train neural networks using low-depth circuits.
- Wong, T. G. (2022). *Introduction to Classical and Quantum Computing*. Published by Rooted Grove, Omaha, Nebraska.