

Week 2: Optimization problems & Rydberg atom arrays

Introduction

Last week, you were able to simulate elements of the seminal work produced from Google’s Sycamore quantum computer on your own (classical) laptops. However, given that you are in the CDL stream, you are probably wondering how one might be able to demonstrate a quantum advantage in a “real-world” setting, such as optimization problems, instead of the simulatability of an “unusual” probability distribution (the Porter-Thomas distribution).

Superconducting-qubit (Sycamore) and trapped-ion devices are considered top contenders for scalable quantum computers, due to their long coherence times and high gate fidelities. However, recently neutral-atom quantum computers are also proving to be competitive due to the ease of control of a large number of programmable qubits [1, 2, 3, 4, 5]. Given the immense scaling solutions required for today’s largest data-driven problems (e.g. supply chain management), it seems likely that neutral-atom quantum computers could be close close to demonstrating a quantum advantage for today’s real problems.

The foundation of today’s neutral-atom quantum computers is comprised of Rydberg atoms [5]. Briefly, Rydberg atoms are highly-excited atoms (e.g. rubidium) that interact with each other on the scale of a few micrometres. A controlled laser pulse can excite an atom into a quantum state with a large principal quantum number (i.e. a Rydberg state) that is quasi-stable. The binary nature of a Rydberg atom’s ground and excited states is analogous to the state of a qubit. Besides ease of control and scalability, the value of Rydberg atoms in the context of solving real-world problems is in the way that Rydberg atoms interact with each other. Their interactions map trivially to a well-known mathematical problem called the Unit-Disk Maximum Independent Set (UD-MIS) problem, which is NP-hard¹.

A host of today’s real-world problems are classified as NP-hard, and it turns out that NP-hard problems can be translated into other NP-hard problems with some negligible overhead. So, we can map a lot of real-world problems to the building blocks of quantum computers! This is precisely what you will be investigating this week: exploring the UD-MIS problem as it pertains to Rydberg atoms, and mapping it to a real-life NP-hard problem. First, let’s look at some of the relevant math so that you can do your tasks!

Modelling Rydberg atoms

Rydberg atoms need to be placed at a physical location. We will strictly look at Rydberg atoms on a *graph* G with vertices (physical Rydberg atom locations) and edges V and E , respectively. With

¹Problems that are NP-hard are not solvable in polynomial time, but trial solutions can be verified to be correct or not in polynomial time.

this, we will look at a Rydberg Hamiltonian of the form

$$\hat{H} = - \sum_{i \in V} \hat{n}_i + \sum_{i < j} \left(\frac{R_b}{r_{ij}} \right)^6 \hat{n}_i \hat{n}_j, \quad (1)$$

where $\hat{n}_i = 1/2(I - \hat{\sigma}_i^z) = |1\rangle\langle 1|_i$ is called an *occupation operator*, R_b is a parameter called the *blockade radius*, and r_{ij} is the distance between Rydberg atoms located at vertices i and j . The computational basis we will be working in is the occupation basis: the eigenstates of \hat{n}_i ,

$$\hat{n}_i |0\rangle_j = 0 \quad (\forall i, j), \quad (2a)$$

and

$$\hat{n}_i |1\rangle_j = \delta_{i,j} |1\rangle_j, \quad (2b)$$

where the state $|0\rangle$ ($|1\rangle$) represents the ground (excited) state of a Rydberg atom.

On observing the form of the Hamiltonian in Eq. 1, we can see that the first term (sum over V) favours all sites being occupied, while the interaction term penalizes occupied pairs. It is precisely this dichotomy that leads us to our next section.

Rydberg atoms and the UD-MIS problem

The MIS problem² is defined as follows [1]:

Let $G = (V, E)$ be a graph with a set of vertices V and edges E , $N = |V|$, and $S = (n_1, \dots, n_N)$ be an N -bit string (i.e. $n_i \in \{0, 1\}$) with Hamming weight $|S| = \sum_{i=1}^N n_i$. The MIS problem is defined as finding

$$\max_{S \in \mathcal{B}} |S|,$$

subject to the constraint that S is an independent set. Here, \mathcal{B} is the set of all possible N -bit strings, and an independent set is defined as a set with mutually non-connected vertices: $n_i = n_j = 1 \Rightarrow (i, j) \notin E$.

Solving this problem consists of finding the largest possible independent set and returning an instance of it. With the additional unit-disk constraint, all vertices of the given graph share an edge except those that are separated by a distance greater than 1. An example of an independent set on a unit-disk graph is shown in Fig. 1.

The ground state of the Hamiltonian

$$\hat{H} = - \sum_{i \in V} \hat{n}_i + u \sum_{i, j \in E} \hat{n}_i \hat{n}_j \quad (3)$$

is precisely the solution to the UD-MIS problem on a graph G^3 . This is eerily similar to Eq. 1, with the difference being in the coefficients in front of the interaction terms. However, to a very

²Not UD-MIS!

³So long as $u > 1$ [1]. You will use $u = 1.35$ for all of your tasks.

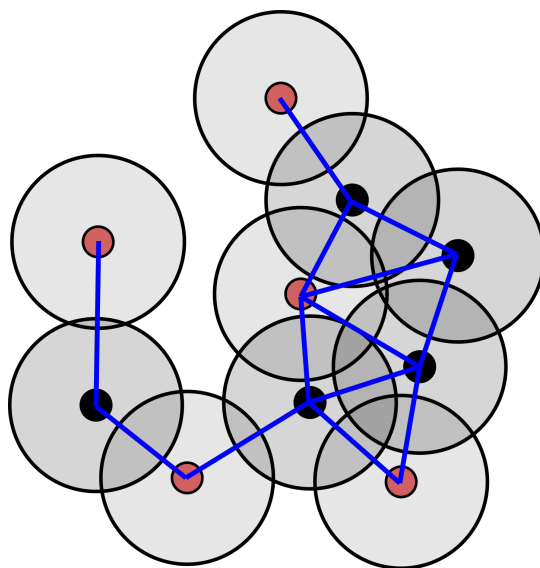


Figure 1: An example of a unit-disk graph G with an instance of the corresponding solution to its UD-MIS problem. Here, blue lines indicate graph edges E , and dots represent vertices V wherein the shaded regions represent unit-radii disks. The red dots correspond to a maximum independent set.

good approximation, it's "similar enough". So, essentially, finding the ground state of a bunch of Rydberg atoms on the same graph G will (approximately) solve the UD-MIS problem.

In all of your tasks, we will be modelling Eq. 3 for simplicity. However, let's just take a moment to appreciate how having a physical neutral-atom quantum computer makes solving the UD-MIS problem so simple. We could just place the Rydberg atoms at the desired vertex locations, and measure their Rydberg occupation! Pretty easy⁴!

Task 1: Simulated Classical Annealing

Let's first try and solve the UD-MIS problem classically. You'll notice that the Hamiltonian in Eq. 3 is diagonal in the Rydberg occupation basis (Eq. 2). So, we can use classical Monte Carlo methods (namely the Metropolis-Hastings algorithm) to simulate this Hamiltonian at finite temperatures. But, we're interested in the *ground* state, not some high-temperature state. That being said, so long as we can simulate the Hamiltonian at a low enough temperature, it should be fine.

In this directory, you will find a Jupyter notebook titled `Task1.ipynb`. Here, there's all the code you will need to solve the UD-MIS problem via classical *simulated annealing*: performing Monte Carlo simulations at lower and lower temperatures to hopefully simulate the ground (zero-temperature) state. The *annealing schedule* is the manner in which the temperature is decreased. We've provided an annealing schedule for you, but **try other annealing schedules to see if you can get a solution to the UD-MIS problem faster (less steps)**.

⁴Obviously, a lot more goes into it. These experiments are highly non-trivial affairs... No free lunch!

Task 2: Quantum Annealing

Let's now try solving the UD-MIS problem quantum-ly. We will employ a quantum annealing (QA)-based approach. Briefly, the goal of QA is to start in an easily-preparable ground state of a Hamiltonian and time evolve this state with a time-dependent Hamiltonian. At the end of the time evolution process, the Hamiltonian should be such that its ground state is what we are after. Consider the time-dependent Hamiltonian

$$\hat{H}(t) = \Omega(t) \sum_{i \in V} \hat{\sigma}_i^x - \delta(t) \sum_{i \in V} \hat{n}_i + u \sum_{i,j \in E} \hat{n}_i \hat{n}_j. \quad (4)$$

Notice that if $\Omega(t=0) = \delta(t=0) = 0$, the ground state of this Hamiltonian is every Rydberg atom in its ground state: $|0\rangle^{\otimes N}$. If $\Omega(t=t_{\text{final}}) = 0$ and $\delta(t=t_{\text{final}}) = 1$, we have Eq. 3. Time-evolving the initial state $|0\rangle^{\otimes N}$ with the Hamiltonian in Eq. 4 by manually choosing how to vary $\Omega(t)$ and $\delta(t)$ will hopefully give us the ground state of the UD-MIS Hamiltonian (Eq. 3). In a nutshell, this is the QA process as it pertains to our problem.

Mathematically and algorithmically, QA looks like the following

$$|\psi(t)\rangle = U(t)U(t-\delta t) \cdots U(t_0 + \delta t)U(t_0)|\psi(t=t_0)\rangle,$$

where $U(t)$ is the time-evolution operator

$$U(t) = \exp\left(-\frac{i}{\hbar}\delta t \hat{H}(t)\right),$$

$\hat{H}(t)$ is given in Eq. 4, and δt is a user-defined increment of time that defines the number of subdivisions in the time interval $t \in [t_0, t_f]$. We will implement this on a quantum circuit simulator package in the Julia programming language called `Yao.jl` [6]⁵. Everything you need is in `run_quantum_annealing.jl`. We have taken inspiration from Ref. [1] for the annealing schedule (see Fig. 2 in this work). The very last line in `run_quantum_annealing.jl` draws samples from the resulting wavefunction $|\psi(t)\rangle$ that has been time-evolved. **Plot the samples on the graph coordinates given in the code and verify that indeed the UD-MIS problem has been solved.**

The resulting time-evolved wavefunction may not *perfectly* be the ground state (i.e. when you sample it, you might get configurations that aren't the solution we desire). Sample the resulting wavefunction several thousand times, and look for the most common bit string. Use this as your solution⁶.

Task 3: A Real Problem

The City of Gotham is looking at putting in new cell phone towers. The possible locations of the cell phone towers are given in Fig. 2. The billionaire Bruce Wayne is funding the project and he loves his money. Therefore, Gotham should only purchase the required number of cell phone towers such that 1) the cell phone tower signal ranges do not overlap⁷, and 2) as much of Gotham City can be within cell signal range.

The possible Gotham City cell phone tower locations are⁸:

⁵Created by our very own Roger Luo!

⁶You might need to run the code a few different times!

⁷Yes, this means some areas of Gotham City won't have cell phone service... "Too bad," said Bruce.

⁸All coordinates are normalized to the cell phone tower signal range.

1. (1.19, 4.25)
2. (2.71, 3.48)
3. (1.19, 3.51)
4. (2, 3.38)
5. (1.12, 2.86)
6. (1.70, 2.42)
7. (2.36, 2.54)
8. (1.52, 1.48)
9. (2.15, 1.54)
10. (2.14, 1.87)
11. (1.72, 0.86)
12. (2.29, 0.87)

1. **Explain why this is a problem that can be easily mapped to the UD-MIS problem.**
A few sentences is all that is needed.
2. **Solve Gotham City's problem.** Use any method you want (i.e. the codes provided in Tasks 1 and 2, or any other UD-MIS solving method), but extra points will be given to groups that demonstrate originality. **Are there multiple solutions?**
3. **Should Bruce pay for a few more cell phone towers to make sure that more of Gotham City has cell phone service?**

Additional Challenges

- Perform any of the tasks with real quantum hardware.
- Compare / benchmark the methods outlined in the Task 1 and 2 codes (if you used a different method in Task 3, consider comparing to that also). You will probably want to make larger graphs to push both of these algorithms to the test. Which method is better (i.e. faster, more efficient)?
- Solve another real-world problem that can be mapped to the UD-MIS problem.

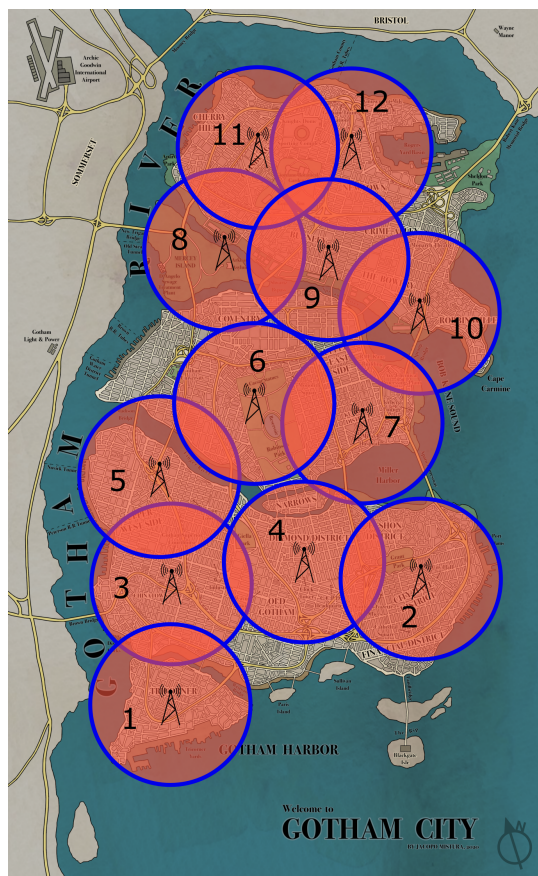


Figure 2: Possible cell phone tower locations in Gotham City.

References

- [1] Michel Fabrice Serret, Bertrand Marchand, and Thomas Ayrat. Solving optimization problems with Rydberg analog quantum computers: Realistic requirements for quantum advantage using noisy simulation and classical benchmarks. November 2020. arXiv: 2006.11190.
- [2] H. Bernien, S. Schwartz, A. Keesling, H. Levine, A. Omran, Hannes Pichler, Soonwon Choi, Alexander S. Zibrov, Manuel Endres, Markus Greiner, Vladan Vuletić, and Mikhail D. Lukin. Probing Many-Body Dynamics on a 51-Atom Quantum Simulator. *Nature*, 551(7682):579–584, November 2017. doi:10.1038/nature24622.
- [3] Sepehr Ebadi, Tout T. Wang, Harry Levine, Alexander Keesling, Giulia Semeghini, Ahmed Omran, Dolev Bluvstein, Rhine Samajdar, Hannes Pichler, Wen Wei Ho, Soonwon Choi, Subir Sachdev, Markus Greiner, Vladan Vuletic, and Mikhail D. Lukin. Quantum Phases of Matter on a 256-Atom Programmable Quantum Simulator. December 2020. arXiv: 2012.12281.
- [4] Loic Henriet, Lucas Beguin, Adrien Signoles, Thierry Lahaye, Antoine Browaeys, Georges-Olivier Reymond, and Christophe Jurczak. Quantum computing with neutral atoms. *Quantum*, 4:327, September 2020. arXiv: 2006.12326. doi:10.22331/q-2020-09-21-327.
- [5] Antoine Browaeys and Thierry Lahaye. Many-body physics with individually controlled rydberg atoms. *Nature Physics*, 16(2):132–142, Feb 2020. doi:10.1038/s41567-019-0733-z.
- [6] Xiu-Zhe Luo, Jin-Guo Liu, Pan Zhang, and Lei Wang. Yao.jl: Extensible, Efficient Framework for Quantum Algorithm Design. *Quantum*, 4:341, October 2020. doi:10.22331/q-2020-10-11-341.