

# 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 simulability of an “odd” probability distribution (the Porter-Thomas distribution).

While superconducting-qubit (Sycamore) and trapped-ion quantum computers present high gate fidelities compared to neutral-atom quantum computers (i.e. neutral-atom quantum computers are more error-prone/noisy) [1], neutral-atom quantum computers are more scalable in terms of the physical number of programmable qubits owing to their ease of experimental control [2, 3, 4, 5]. Logically, if we have access to larger quantum computers, we can foreseeably demonstrate a quantum advantage for this very reason. Not only this, but given the immense scaling solutions required for today’s largest data-driven problems (e.g. supply chain management), it seems more likely that neutral-atom quantum computers are closer to demonstrating a quantum advantage for today’s real problems.

The foundation of today’s neutral-atom quantum computers comprise of Rydberg atoms [5]. Briefly, Rydberg atoms (e.g. rubidium) are large atoms that interact with each other on the scale of a few micrometres. A controlled laser pulse can then excite a Rydberg atom into a quantum state with a large principal quantum number that is quasi-stable. The binary nature of a Rydberg atom’s ground and excited states gives analogous mappings to qubits. 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 with each other map trivially to a well-known mathematical problem called the Unit-Disk Maximum Independent Set (UD-MIS) problem, which is NP-hard<sup>1</sup>.

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 exploring this week: exploring the UD-MIS problem as it pertains to Rydberg atoms, and mapping the UD-MIS problem to a real-life NP-hard problem. First, let’s look at some math so that you can do your tasks!

## Modelling Rydberg atoms

Rydberg atoms need to be given 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

---

<sup>1</sup>Problems 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 sum over  $V$  term 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 problem<sup>2</sup> 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 the solution to the constraint

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

such 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 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  $G$ <sup>3</sup>. This is eerily similar to Eq. 1, with the difference being in the coefficients in front of the interaction terms. However, to a very

---

<sup>2</sup>Not UD-MIS!

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

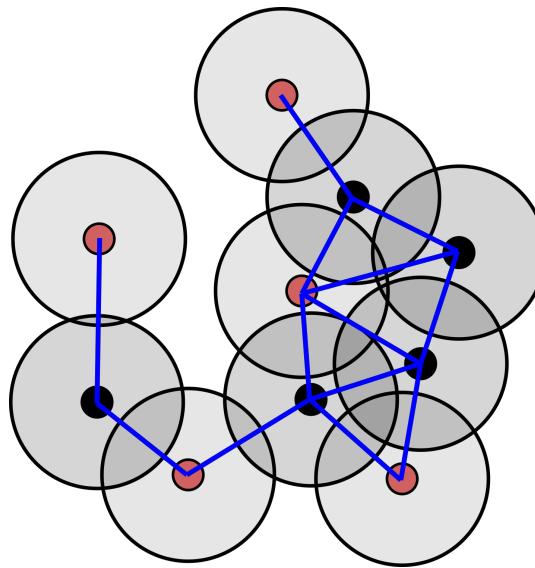


Figure 1: An example of a unit-disk graph  $G$  with an instance of the corresponding solution to its UD-MIS problem. Here, blue lines indicate graph edges  $E$ , and dots represent vertices  $V$  wherein the shaded regions represent unit-radii disks. The red dots correspond to a maximum independent 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 easy<sup>4</sup>!

## Task 1: Classical Solver

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 a finite temperature. 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)**.

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

## Task 2: (Simulated) Quantum Solver

Let's now try solving the UD-MIS problem quantum-ly in the following way. Consider an arbitrary wavefunction  $|\psi_0\rangle$  written in the Rydberg occupation basis,

$$|\psi_0\rangle = \sum_{S \in \mathcal{B}} c_S |S\rangle. \quad (4)$$

Letting  $|\psi_0\rangle$  time-evolve for a length of time  $t$  according to the time-dependent Schrodinger equation,

$$|\psi(t)\rangle = \exp\left(-\frac{i}{\hbar}t\hat{H}\right) |\psi_0\rangle, \quad (5)$$

and using a well-known result of quantum mechanics stating that so long as we time-evolve for long enough and slowly enough, we will arrive at the ground state of the Hamiltonian (in our case, Eq. 3). Let's look at performing the time-evolution exactly with a Julia package called `Yao.jl` [6]<sup>5</sup>. Everything you need to solve this problem is in `run_time_evolution.jl`. The very last line draws samples from the resulting wavefunction that has been time-evolved. **Plot the samples on the graph coordinates given in the code and verify that indeed the UD-MIS has been solved.**

## Task 3: A Real Problem

The city of Gotham is looking at putting in new cell phone towers in the city. 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 overlap<sup>6</sup>, and 2) as much of Gotham city can be within cell signal range.

The possible Gotham City cell phone tower locations are<sup>7</sup>:

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)

---

<sup>5</sup>Created by our very own Roger Luo!

<sup>6</sup>Yes, this means some areas of Gotham city won't have cell phone service... "Too bad," said Bruce.

<sup>7</sup>All coordinates are normalized to the cell phone tower signal range.

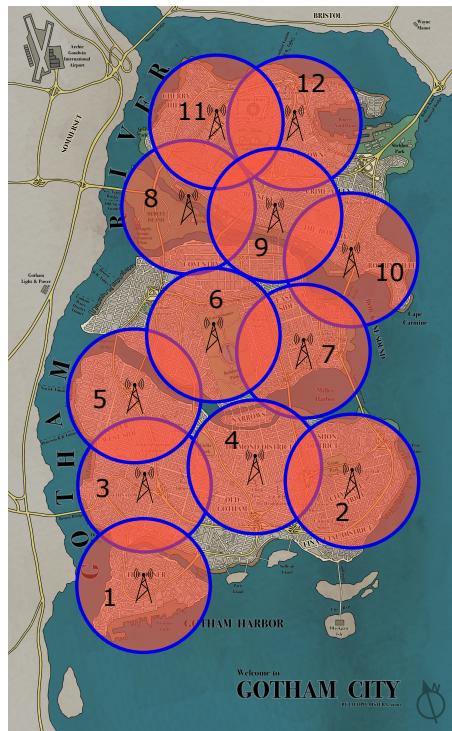


Figure 2: Possible cell phone tower locations in Gotham City.

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.
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.

## References

- [1] Michel Fabrice Serret, Bertrand Marchand, and Thomas Ayral. Solving optimization problems with Rydberg analog quantum computers: Realistic requirements for quantum advantage using noisy simulation and classical benchmarks. *arXiv:2006.11190 [cond-mat, physics:quant-ph]*, November 2020. arXiv: 2006.11190. URL: <http://arxiv.org/abs/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. URL: <https://www.nature.com/articles/nature24622>, 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. *arXiv:2012.12281 [cond-mat, physics:physics, physics:quant-ph]*, December 2020. URL: <http://arxiv.org/abs/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. URL: <http://arxiv.org/abs/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. arXiv: 1912.10877. URL: <http://arxiv.org/abs/1912.10877>, doi:10.22331/q-2020-10-11-341.