QuEra iQuHack 2024 Design Challenge
(Masarah Ahmedhussain, Elizabeth Foster, Avery Hanna, Noah Stiegler, Temazulu Zulu)

The Problem: QuEra's challenge asked us to consider the Defective King's Lattice, a graph in which vertices are positioned in a square lattice with a few vertices removed randomly, and edges exist up to 8 surrounding vertices. We can then find the Maximum Independent Set on the resulting graph. The Maximum Independent Set problem is an NP-hard problem with an exponential classical algorithm. However, there is a proven potential for a quantum speedup. This problem expands as we adjust what radius the edges can connect to. The traditional king's lattice problem only allows the king to move one space in any direction, like in a chess game. This means that none of the immediately adjacent nodes to a king may be in the Maximum Independent Set. What if we extended the moves of the king to 2 squares in any direction? What if we shrunk this radius to exclude the corners? Previous studies have determined that a radius that includes a range of 2-4 vertices experiences the peak in the difficulty of solving this problem. QuEra's Aquila quantum architecture provided a unique environment to test the problem.

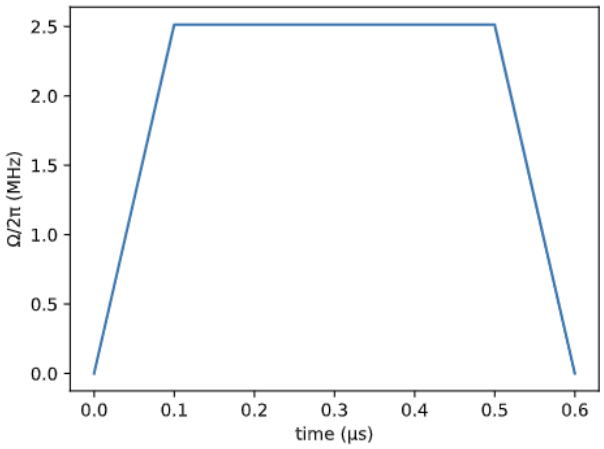
Goal: Figuring out an optimization function that uses the adiabatic approach that is even a fraction larger than the known one, which was the King's value.

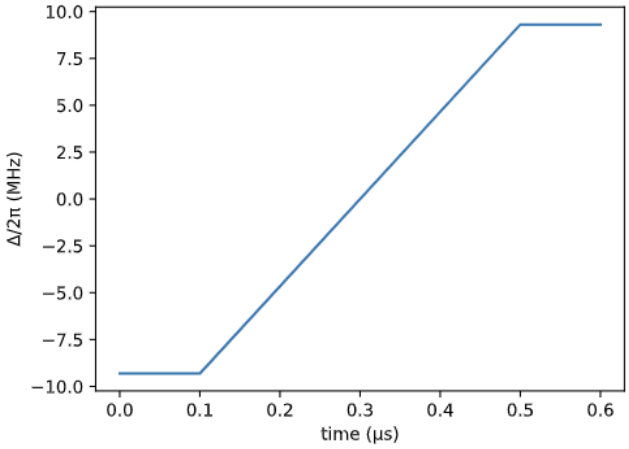
Approach/Theory: We spent a huge part trying to understand the problem, the quantum physics and math behind it, having no prior knowledge. We worked through this theory:

- How the hardware is running - in this case we have a laser/set of lasers that excite Rydberg atoms from the ground state into Rydberg states. The excitation frequency (Rabi frequency) is Omega and is controlled by the laser power. For delta, it is the change in energy that makes sure that no two atoms that are in their superposition states are too close to each other. The goal is to have the atoms in a superposition, where there is a distribution between excited and ground state, and they are not all completely excited or in the ground state. There exists a Rydberg constraint, where no two atoms are excited to the Rydberg state if they are too close to each other. We understood the problem and figured out that the constraints to the Rydberg radius were the Rabi frequency, detuning, and the laser phase.
- We found a relationship between the Blockade radius, a, Omega, and Delta $R_b = \frac{C_6}{\sqrt{\Omega^2+\Delta^2}}$, $r = R_b a$, and $\Delta_{final} = (C_6 \frac{a}{r})^6 = (C_6 \frac{a}{(\frac{R_b}{a})})^6 = (\frac{C_6}{R_b})^6$ because Omega = 0 when the state is measured and so doesn't affect the blockade radius. We learned that the blockade radius isn't a sharp cutoff for interaction, and decided to choose a value that was the geometric mean of the minimum and maximum radii it could take while still including the atoms we wanted and excluding the atoms we didn't. We realized that the smaller a, the larger the number of atoms that could be within each others' blockade radius, so chose our smallest value of a=4um. Within that constraint, the final blockade radius would be affected by the final delta value, which we would slowly ramp up to (so all atoms start in the ground state

and try to be excited unless they are prevented by the blockade interaction energy). We also knew that Omega should be less than Delta because we wanted the annealing process to adjust adiabatically, evolving slowly enough in parameters that the state ends up in the superposition we want. Because Omega has to go to 0 at the end of the measurement, if it's an order of magnitude larger than Delta, then R_b would change rapidly right before measurement, muddying our results. So, in other words, we needed to find a good trade-off between increasing the connectivity of the King's Lattice and the quality of the annealing process. If we wanted connectivity to be higher, we wanted R_b to be higher, however this would decrease Delta (and because we took Omega to be bounded by Delta, also Omega). The smaller Omega and Delta, the less the state was pushed into a superposition and the less it was encouraged to evolve into an MIS. We knew that different Omega and Delta time series profiles might have yielded better results, but didn't have time to investigate them thoroughly, so we mostly stuck with a trapezoidal Omega profile (where it ramped up to 'jumble' the qubits) and a linearly increasing Delta (slowly inflating the 'balloons' of the blockade radii of each atom to force them into the higher state and to compete with each other).

- Our process for finding the Delta_final_max parameter value was:
    - Aim for connectivity r = R_b/a. Find R_b via $\sqrt{R_{min}R_{max}}$
    - $\Delta_{max} = \frac{C_6}{(R_b)^6}$
    - Example Omega and Delta traces





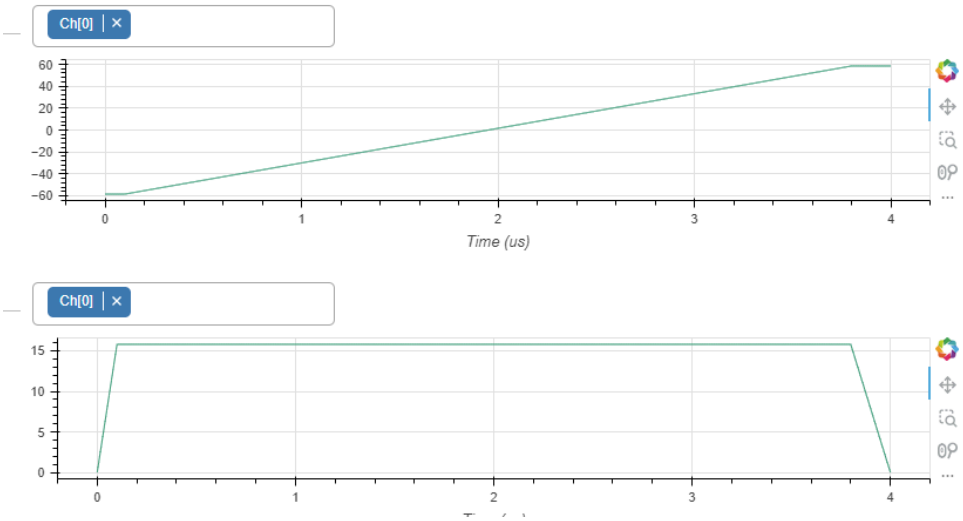
- We created a parameter space we would have been able to search through with optimized parameter traces in order to find at what values of delta_max_final the probability of aquila

a (um)		Final Detuning Frequency (rad/us) for different unit disk radii				
	sqrt(2)	2	sqrt(5)	2sqrt(2)	3	r (Rb/a)
	5.656854249	8	8.94427191	11.3137085	12	Radius
4	58.47959525	14.79429741	5.230574011	2.165910935	1.549799707	
4.25	40.64720855	10.28302076	3.635596857	1.505452169	1.077213883	
4.5	28.84624072	7.297585809	2.580086206	1.068379286	0.764469987	
4.75	20.85463765	5.275852379	1.865295497	0.7723939869	0.5526801473	
5	15.33007502	3.8782363	1.371163593	0.5677805562	0.4062706944	
5.25	11.43953801	2.893999639	1.023183385	0.4236865929	0.3031654473	

returning a likely MIS solution became too small to be detectable:

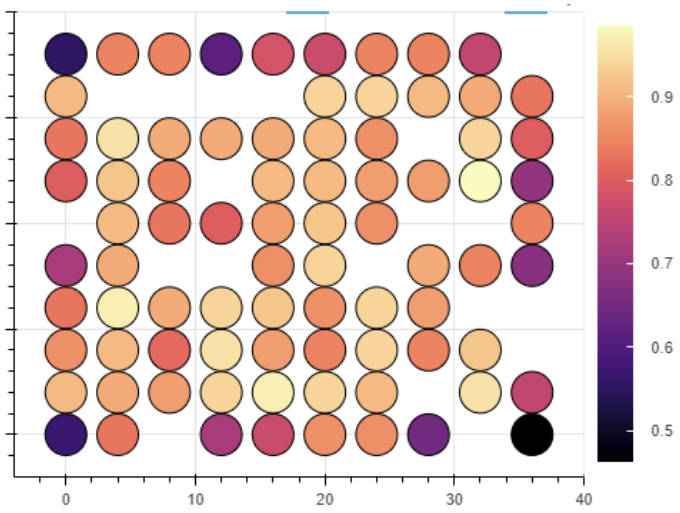
- How that translates to software: We followed through the bloqade julia package on trying to create simulations. We started with some default parameter traces from the example notebooks. We figured we would optimize them later if we had more time. We figured out the parameter constraints where omega should be less than delta, a should be greater than or equal to 4 and the blockade radius should be bigger than or equal to a (and ideally up to 3a). We stuck with a value of a= 4, then we tried different methods to calculate the Rydberg radius. We ran a bunch of simulations on 4x4 systems, but found that for larger interaction radii that the graphs became trivial. We tried to increase the size of the matrix, but the classical quantum simulator took too long to run, so we attempted to run our technique on the quantum processor.
    - Apologies - we struggled for a long time to understand the problem, the software, and explore, so our in-code documentation isn't as detailed and explanative as we'd liked
- We chose the adiabatic approach for our quantum simulator because we understood that we need to slowly ramp up the Rydberg atoms' omega from the ground state. It also was the most intuitive and time saving in terms of understanding and trying to implement.

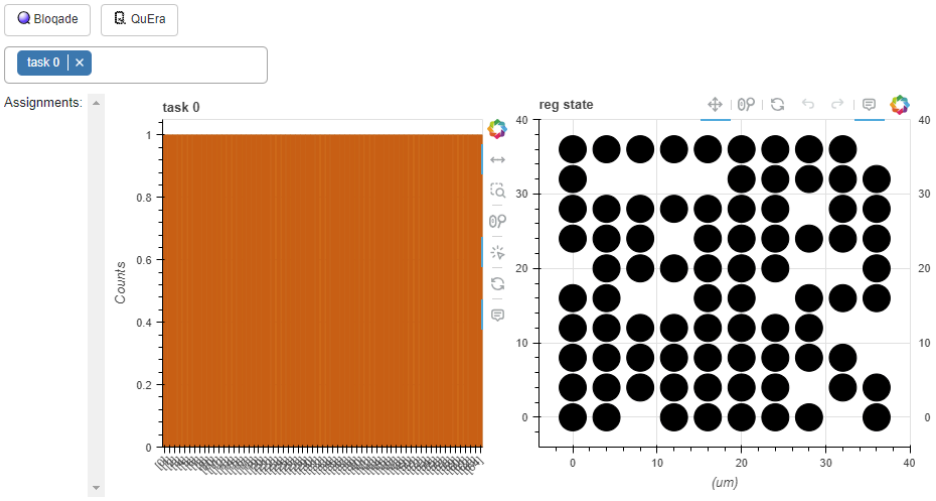
- In terms of results, we explored many classical adiabatic simulations on 4x4 lattices (as anything larger wouldn't run through Bloqade). We tried using subspaces to increase the size of our lattice, but realized they were creating a bias in our results (by enforcing that the blockade radius was a certain size). However, we weren't sure if our parameter values (omega and delta) would create the blockade radius we were expecting, so we didn't want to enforce it with the subspaces. We found that at the lower delta and omega values associated with higher connectivity (even just beyond the next nearest neighbor king's lattice to a value of r (R_b/a) = 2), omega and delta were dropped so low (with our time series profiles) that the results wouldn't produce MIS solutions. We tried running two experiments on the quantum computer to check if our simulations and assumptions about R_b given omega and delta were correct, but given the device error, we weren't sure how to read the results we got back. We created a histogram which told us how many excited states were in each solution, and found that there was a peak around a certain value for a given lattice and R_b, and worked to find the classical MIS solutions to the same lattice to compare MIS solution set size to the histogram peak. Given that we struggled with determining if our R_b calculation was accurate and producing the connectivity we wanted, we weren't confident enough in our solutions to optimize our Omega and Delta parameters.
- We ran a 100-shot trial on Aquila with a (10x10) lattice and a delta_max_final value we



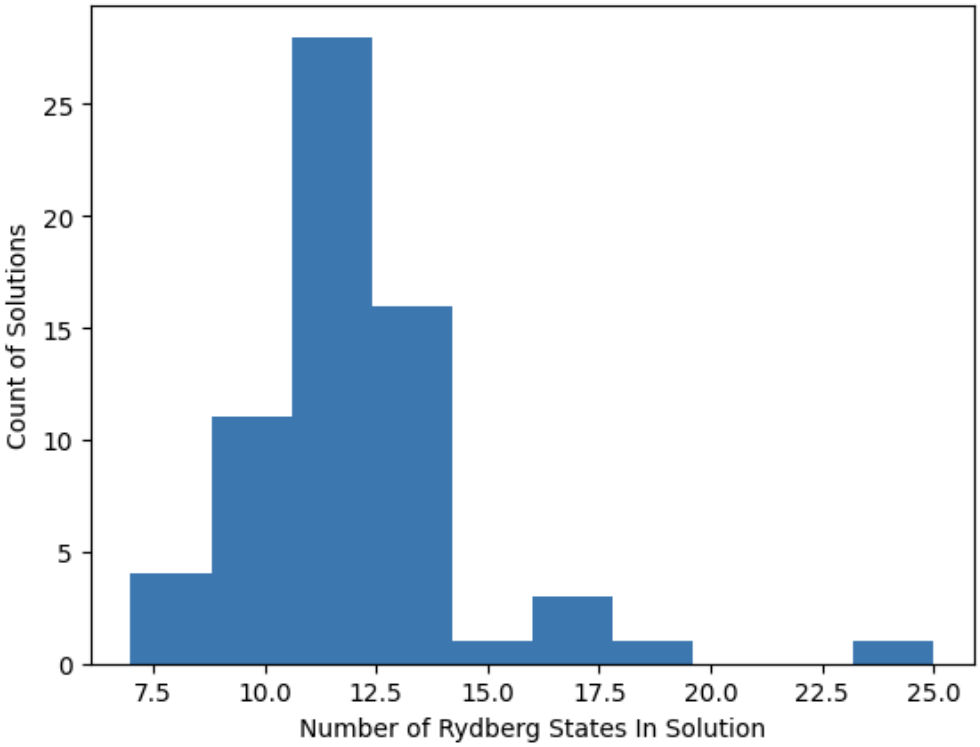
calculated to be associated with an r=2 (14.79 rad/us)

  - Parameter traces for omega and delta for this test -

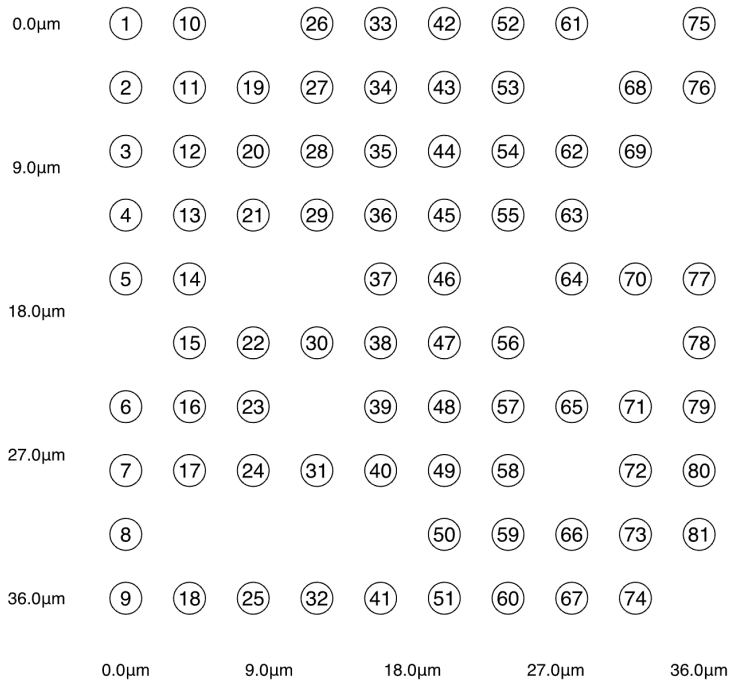


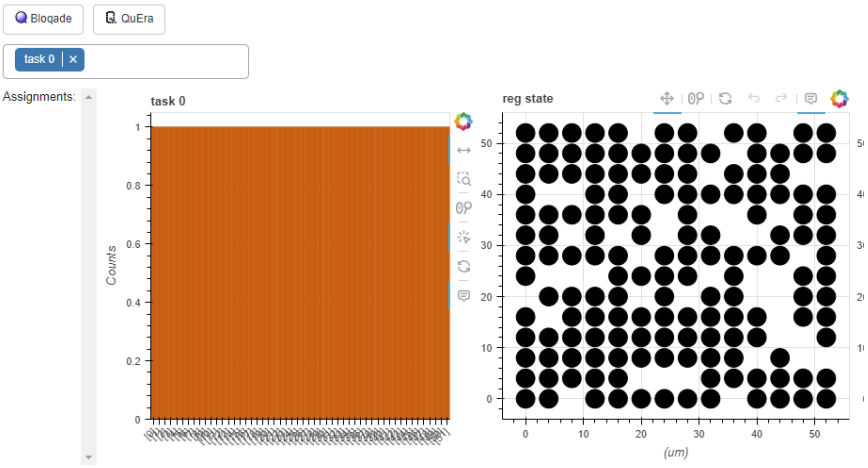


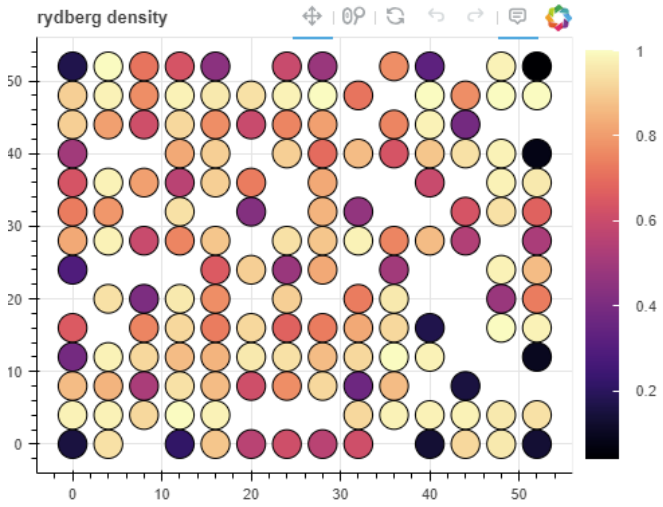
-
-
- Results of the (10x10) r=2 run on Aquila



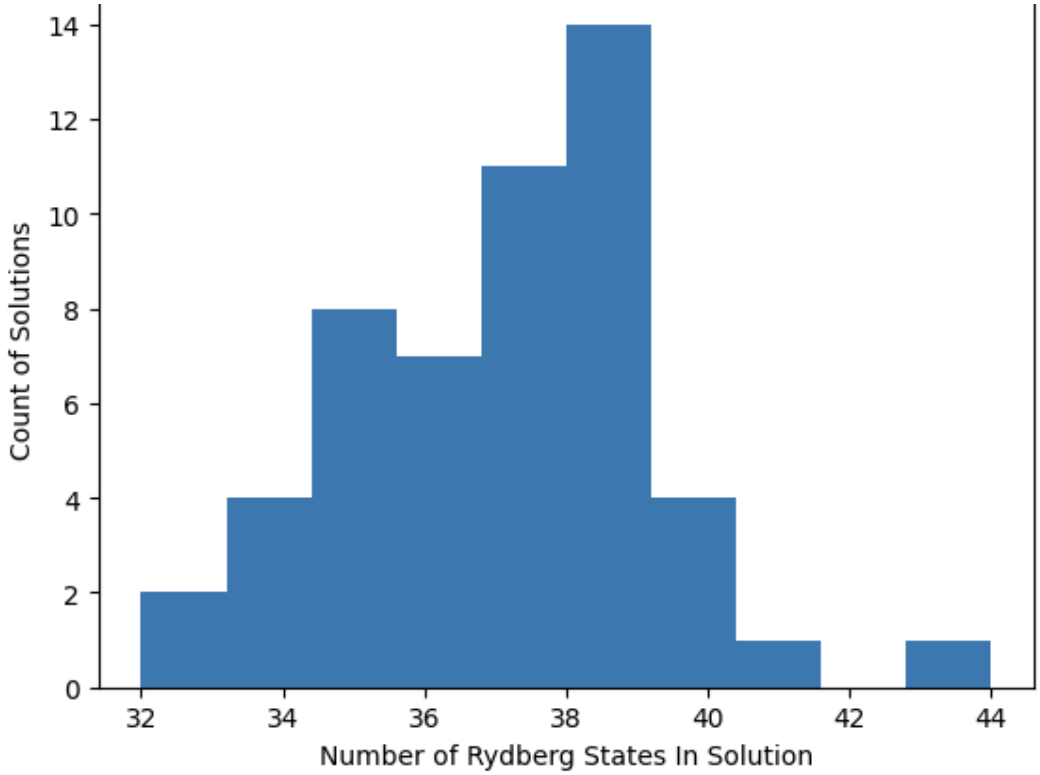
- Histogram of number of excited atoms in solutions returned from Aquila. Peak noticeable at 10-12 excited states, which we hope would correspond to the classical MIS solution set size
- Running classical case, we see there are actually 144 independent sets of the max independent set size of 20



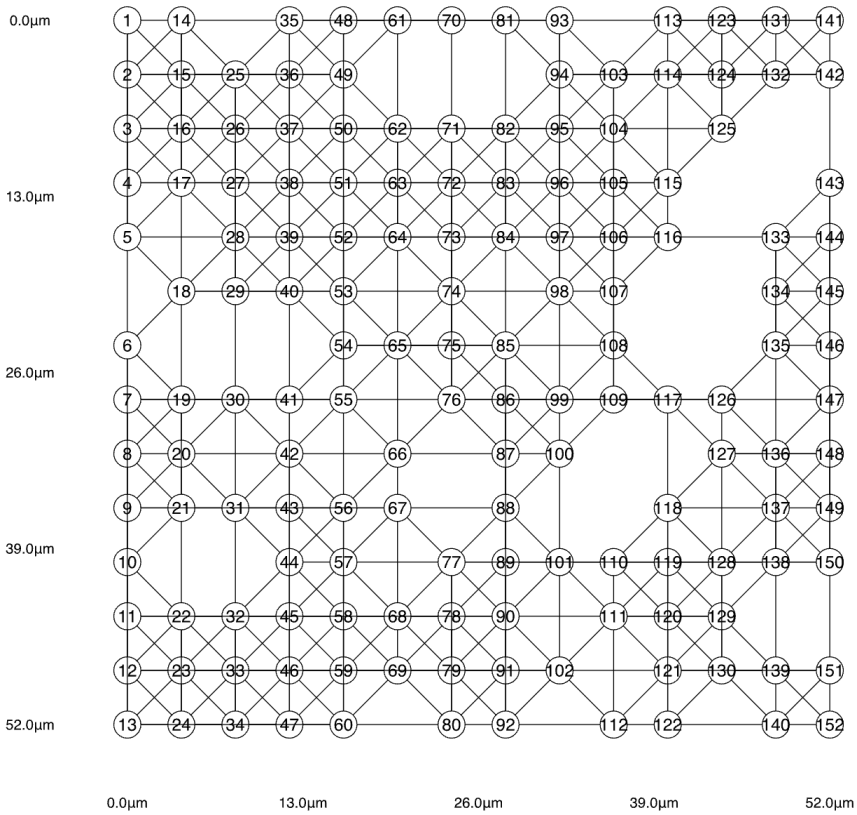
- Ran (14x14) r=sqrt2 simulation on aquila with Delta_final_max = 58.48 (rad/us)
  - 

  - 

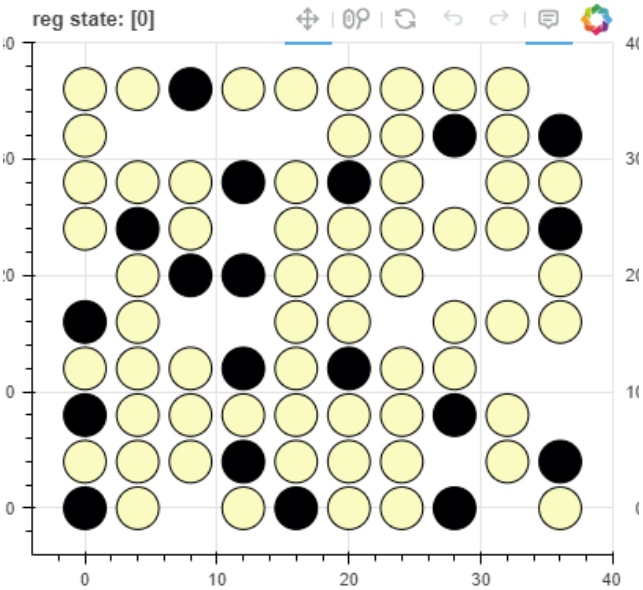
  - Results of (14x14) r=sqrt2 aquila simulation. Note in density that 0 is an excited state



-
  - Results of the (14x14) run on aquila, note the histogram peak around 38-40 excited states
  - Compare this with the classical results that report the size of the maximum independent set as 47



- We noted that there were some non-MIS states in our Aquila results (which could be a result of quantum fluctuations, imperfect quantum simulation, imperfect readout, bad parameter choices). For example, two excited states next to each other. If wed had more time, we would have been able to remove these solutions.



  - Example of non-independent set solution from Aquila. Notice two excited states next to each other

- We struggled with this because the Aquila tasks were sent in Python using the Bloqade python sdk, but our tools for analyzing the graphs were in Bloqade Julia. Just towards the end of the hackathon, we found a fairly manual system for converting the atom positions stored in the Aquila result json into a julia graph, and were barely able to analyze whether the 'actual' solutions found by the classical MIS solvers reflected the peaks of the histograms of number of excited atoms in our quantum results
    - For the 10x10 n=2 matrix, the peak of our histogram suggested an MIS solution with 12-13 nodes, but our classical MIS solutions found an MIS set size of 20
    - For the 14x14 n=sqrt2 matrix, the peak of our histogram suggested an MIS solution with ~40 nodes, but our classical MIS solutions found a set size of 47
- We would hope that with more optimization, our group could have shown an n=2 or n=2sqrt2 large matrix solution having a fairly accurate and quick solution on the quantum hardware. The main advantage of the quantum hardware is that a larger matrix doesn't take more time to solve, so if a solution works well with a 10x10 lattice, we'd hope that it would work well with an arbitrarily large matrix

Business: There are multiple industry applications for the optimized MIS problem on a quantum computer and the most useful we came up with -

- Antenna applications: Figuring out where to place antennas so that they reach the maximum number of people while not overlapping to cause signal interferences. This is key in urban cities where there is a high enough population density to cause slow networks/interference. This would lead to reliable communication lines and some cost reduction benefit to telecommunications companies. Practically, we would have vertices that represent each antenna, and two vertices may share an edge if two communication lines via a signal are communicating. Then the unit graph radius would represent the signal coverage, which is then the constraint. This is a relevant case that would require quantum simulations given that there could exist a very large number of antennas in cities and we would want to figure the optimal combination such that it provides an efficient distribution.
- Healthcare: For example, in a country where there are few medical drug distribution warehouses, which are mostly located in urban areas, we can try to figure out where to build other distribution points closer to rural areas in a way that serves the maximum number of residents, arrives in a timely manner and is still cost-effective.
- Dating mixer: As a fun application of this algorithm, we can consider planning a dating mixer for all of our friends (we won't consider ourselves and our personal connections in this structure). We don't want to have any exes together in the party so we map out our friend group with our friends as nodes and edges representing past relationships. Given we want a large-scale party, to choose our guestlist we take the maximum independent set of the graph.