q.

Corey Ferrier, Avneesh Verma, Dhruv Srinivasan, Jade LeSchack, Jyoti Rani

Problem:

The goal of this year's QuEra challenge is to find the maximally independent sets in a particular UDE (Unitary Disk Graph) of our choice.

Approach and Implementation

Graph Approach

We first approached the problem by trying to consider which graphs we could create that would produce the largest maximally independent set in comparison to the total number of nodes in the graph. When running an assortment of different graph shapes, we determined that the shape of the graph we pick could drastically change our overall results.

To address this, we followed two approaches. The first approach was modifying the kings_graph() function to work a higher number of qubits and density, and cycling through different, random graph types. The second was through physically iterating through different graph designs, and comparing their respective outputs. **We arrived at the conclusion that the largest maximal independent set can contain at most half the amount of nodes present in the graph.** Because having an edge between nodes means that only one of two nodes on that edge can be apart of the independent set, we focused on graphs that had fewer edges per node which led us to simply a *straight line of nodes*. We had to curve this line in order to fit within the bounds of the qpu, maintaining the minimum spacing to most densely pack and array of nodes onto the processor. Additionally, these patterns are of interest to run on quantum computers, due to them being structured as **1D Materials, making it exciting from an applications perspective to optimize for.**

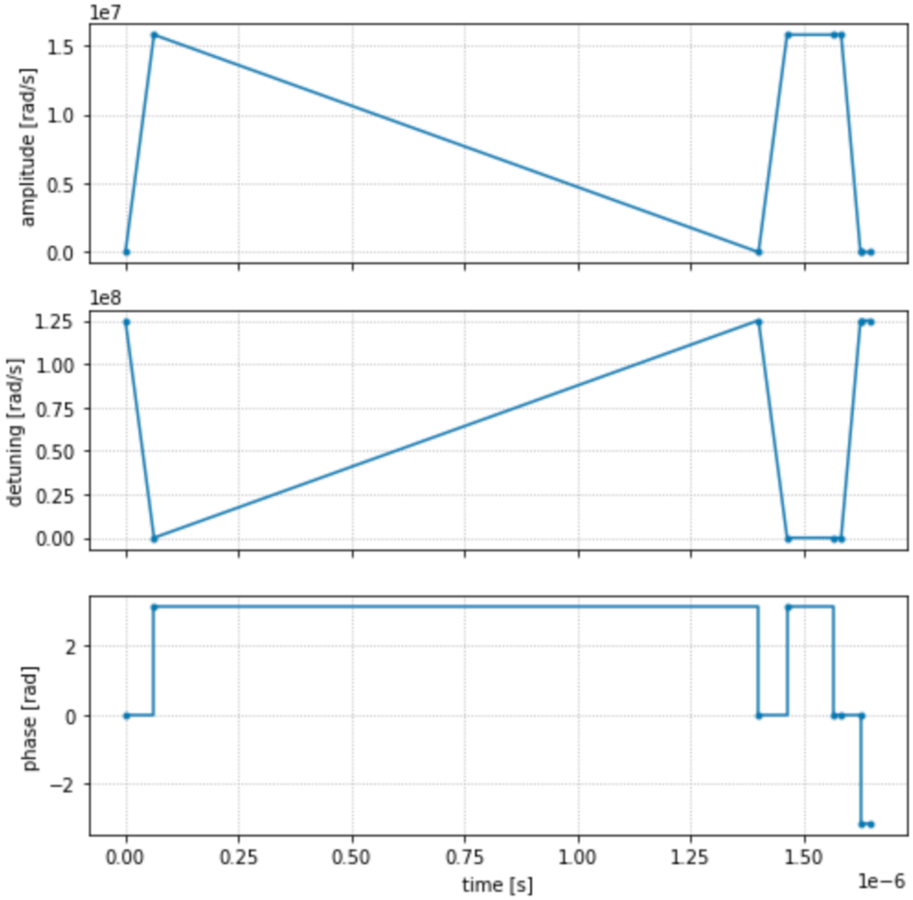
State Preperations and Optimizers Approach

We then focused on trying to optimize the Adiabatic step by looking at the omega, Rabi frequency, and detuning values. We first tried adjusting these values by increasing and decreasing each in isolation and observing the effect it had on the probability of the simulator rendering the correct number for its MIS size, on a 4X4 node grid with 70% density. We then attempted to pulse omega and delta in different ways to observe any other effects that it would have, given the bounds outlined by the machine when importing the qpu. The probability of success would drastically decrease after the aforementioned changes. That is, the algorithm would confidently affirm a fewer number of maximally independent sets for a solution than was correct.

Through Bloqade documentation and picking the mind's of those at QuEra, we began to research how the mathematical expression for each adjustable term in the Hamiltonian related to the qualitative impact when the machine was run, to better inform our parametrization problems. At this stage, we determined to research other optimization algorithms; specifically, Floquet Theorem Models, Counterdiabatics, QAOA, and an alternate approach to the adiabatic methodology with neutral atoms.

Due to limited time constraints, decided to invest time exploring QAOA and the alternate adiabatic approach. The rationale for this was due to its robust documentation in Bloqade and Julia, though literature implementing QAOA using Python on the Aquila was limited. We consulted the “Quantum Optimization for Maximum Independent Set Using Rydberg Atom Arrays” (Pitcher et. al) paper for the alternative adiabatic approach.

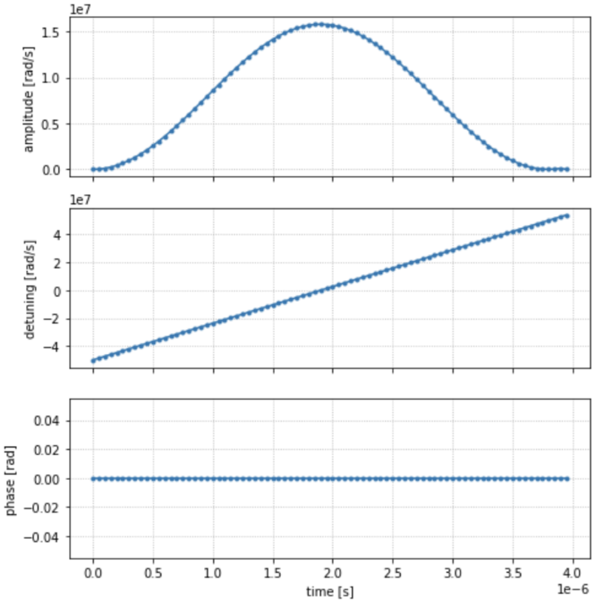
With QAOA, the rabi and detuning terms (often called the mixer and cost hamiltonians in QAOA) are repeatedly switched on and off, in an alternating fashion. While the rabi term is on, the detuning term is off, and while the rabi term is off, the detuning term is on. The duration of each pulse is a parametrized, and optimized by the Scipy COBYLA optimizer.



QAOA posed many challenges. The optimizer would often fail to converge on parameters that satisfied the two constraints imposed:

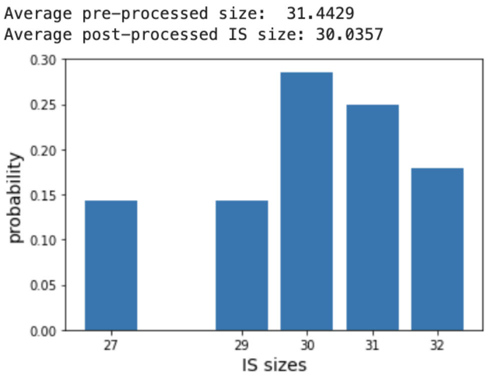
1. All parameters must be positive (since they represent a duration of time)
2. The total time must not exceed the maximum allowable runtime of the QPU

For the Pitchet et. al adiabatic process, we manipulated three parameters of the hamiltonian to create these curves modeled from the associated equations found in the paper:

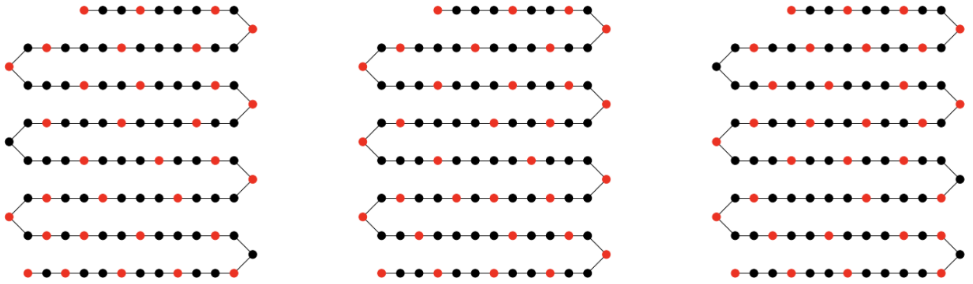


$$\Delta(t) = \Delta_0(2t/T - 1), \quad \Omega(t) = \Omega_0 \sin^2(\pi t/T)$$

Applying these parameters to the model found in run 4 below, we found that there was negligible difference in the output results, performing marginally worse.



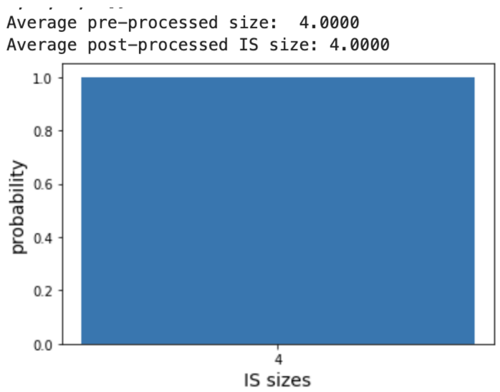
```
Average pre-processed size:  31.4429
Average post-processed IS size: 30.0357
```



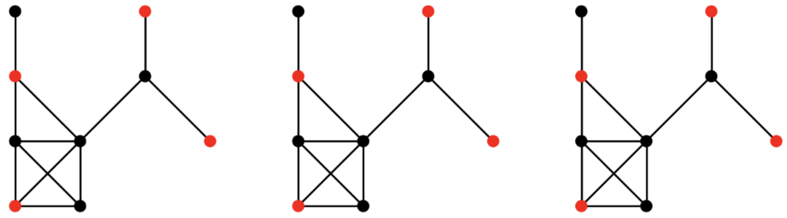
Ultimately, ***we determined that the original adiabatic algorithm was more effective than QAOA at converging to the solution.*** One of the chief challenges was our QAOA implementation accidentally converging on local maxima as opposed to the correct solution (e.g., QAOA would affirm that three MIS was the solution with 100% certainty in the simulator when the correct answer was four for a particular graph). **As such, the solutions we show below are the best solutions we converged on, which came exclusively from the adiabatic process.** Our results when using the quantum computer are on par to those in the examples provided, despite only running on 70 shots compared to the example's 1000 shots.

Notable Results

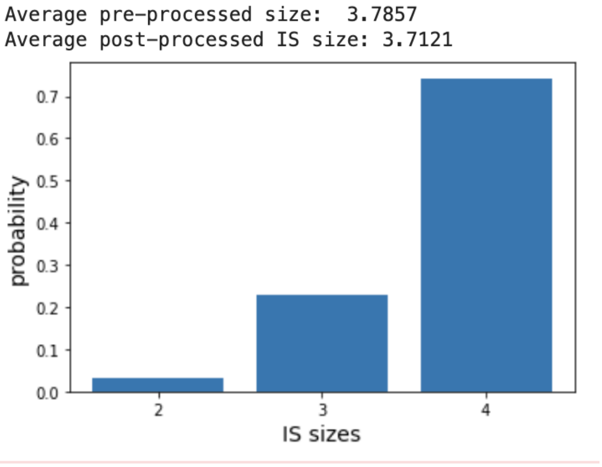
1) 9 Node Graph, 70 Shots



```
Average pre-processed size:  4.0000
Average post-processed IS size: 4.0000
```



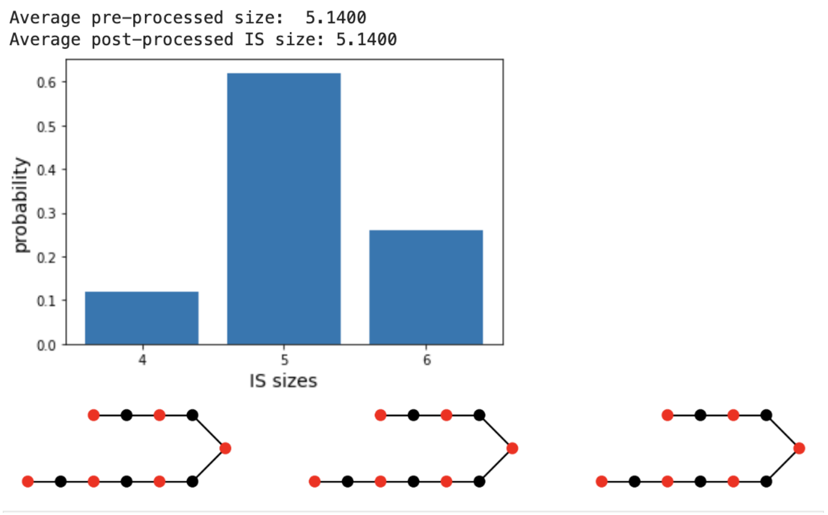
Classical Simulation,100% Convergence on 4 MIS



```
Average pre-processed size:  3.7857
Average post-processed IS size: 3.7121
```

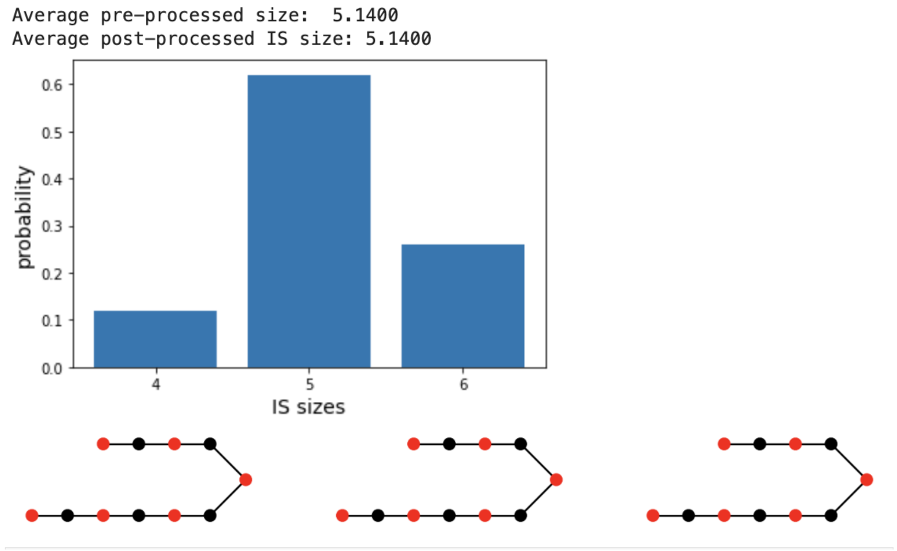
Quantum Computer Result ~73% Convergence

2) 11 Node "Snake", 70 Shots



```
Average pre-processed size:  5.1400
Average post-processed IS size: 5.1400
```

Classical Simulation, affirms 5 as largest probability of MIS, when solution is 6

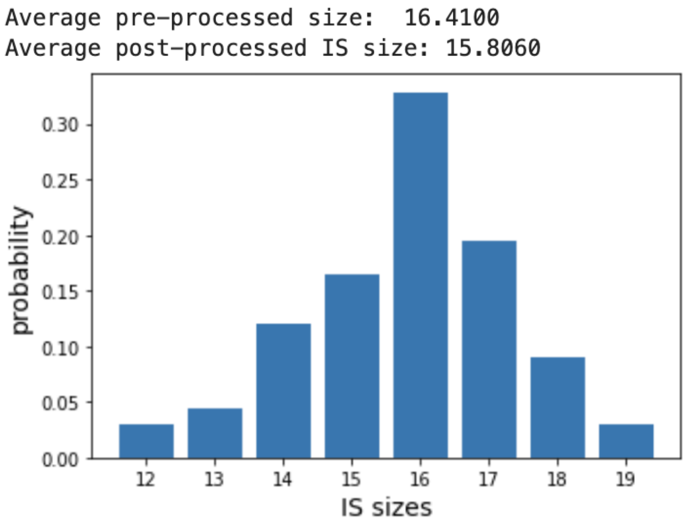


```
Average pre-processed size:  5.1400
Average post-processed IS size: 5.1400
```

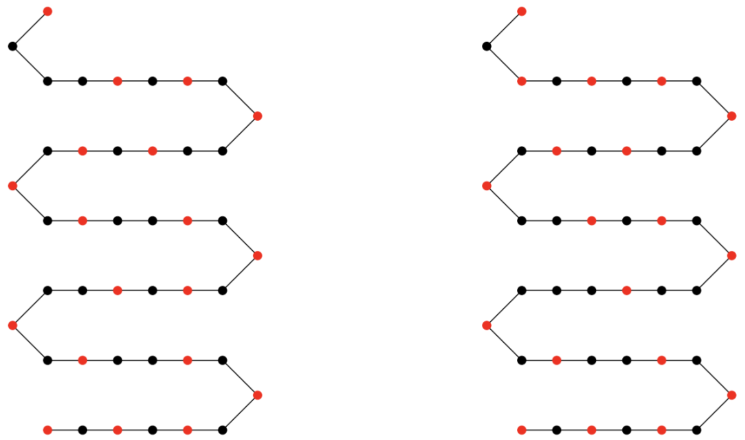
Quantum Computer result, affirms the same as the simulator

The following examples cannot be simulated classically

3) 41 Node "Snake", 70 Shots

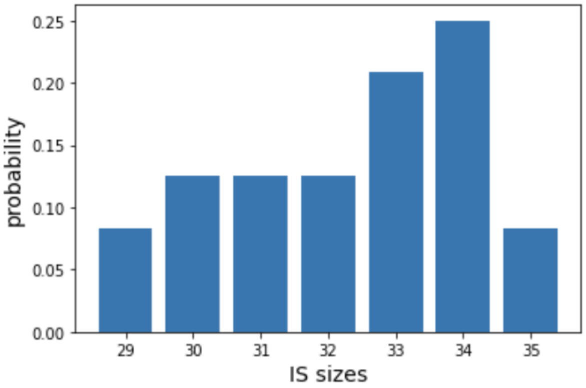


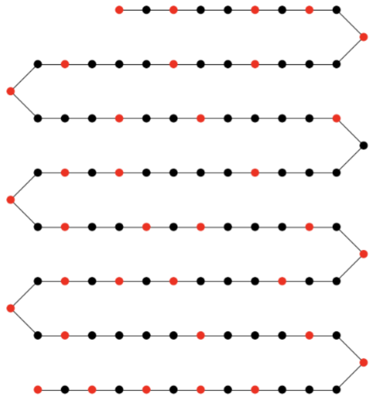
```
Average pre-processed size:  16.4100
Average post-processed IS size: 15.8060
```

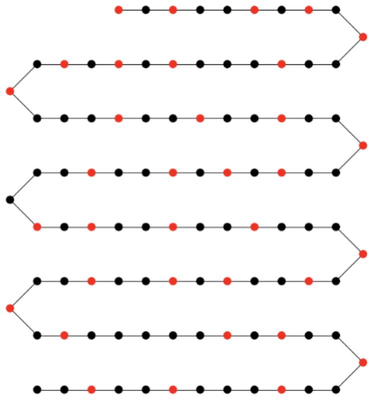


16 MIS found

4) [2nd Best] 100 Node “Snake”, 70 Shots

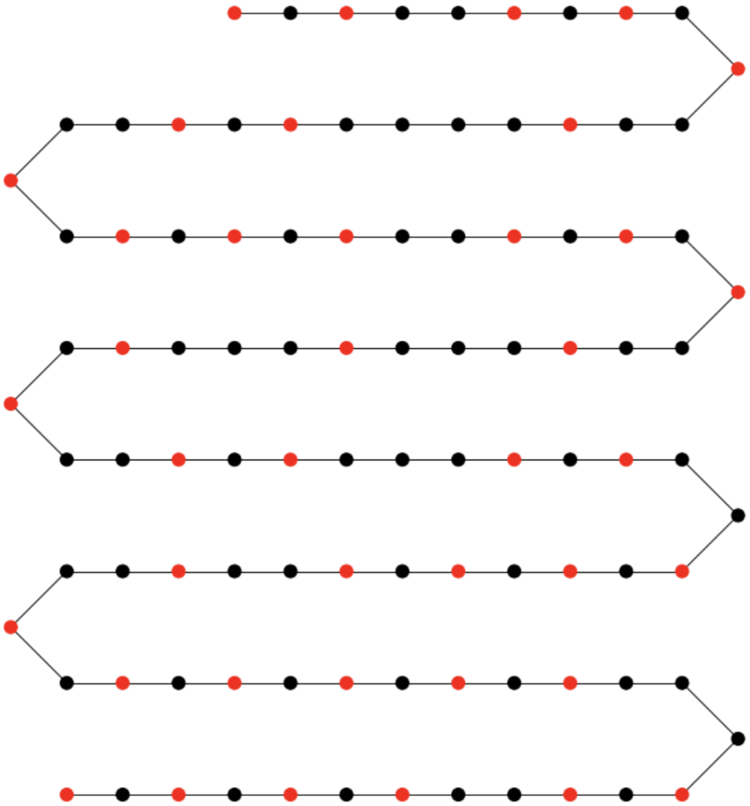




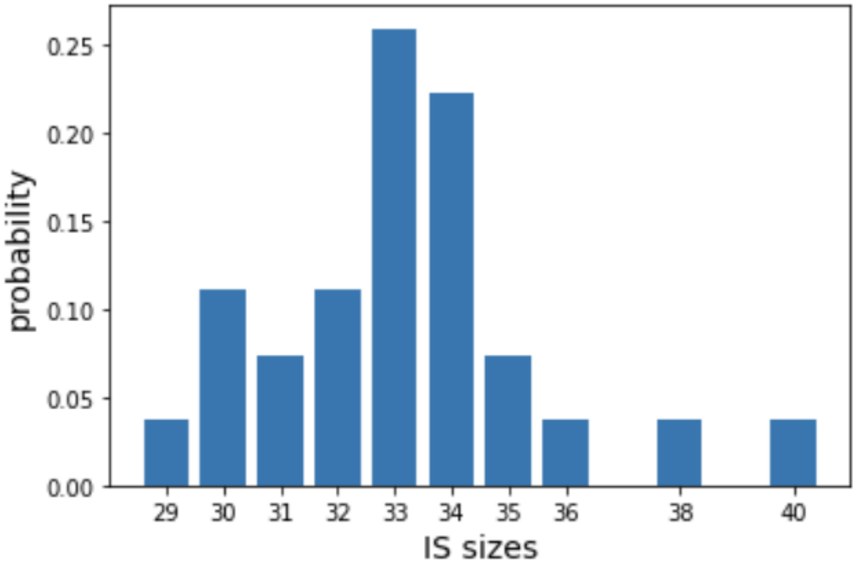


34 MIS Found

5) [1st Best] 100 Node “Snake”, 70 shots

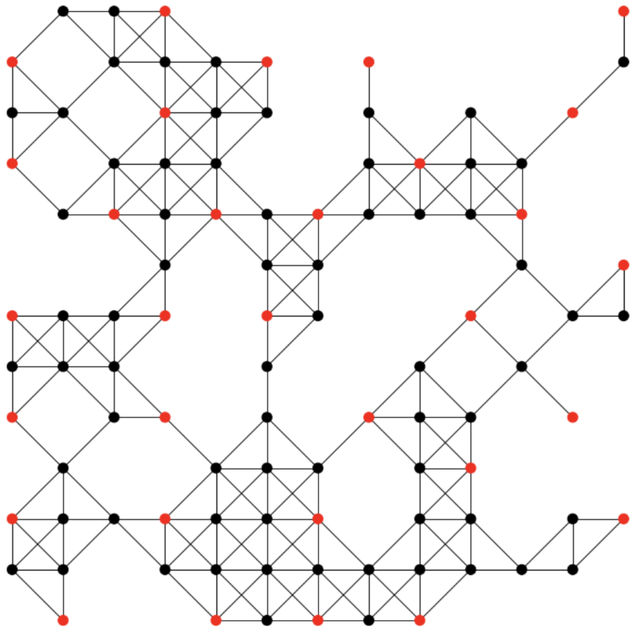


```
Average pre-processed size:  34.9571
Average post-processed IS size: 33.1852
```

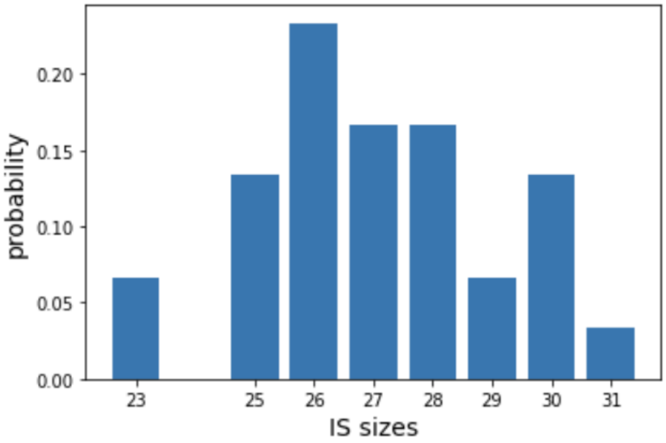


Highest MIS: 40

6) 100 Node “Organic” Shape



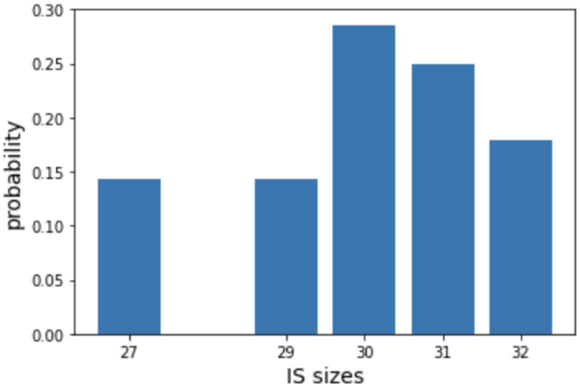
```
Average pre-processed size:  28.5714
Average post-processed IS size: 27.0667
```



31 MIS Found

7) 100 Node Snake with Pitcher et. al adiabatic evolution, 70 shots, smaller blockade radius

```
Average pre-processed size:  31.4429
Average post-processed IS size: 30.0357
```


probability
IS sizes