FIRE: THE FIRST-YEAR INNOVATION & RESEARCH EXPERIENCE

QUANTUM MACHINE LEARNING

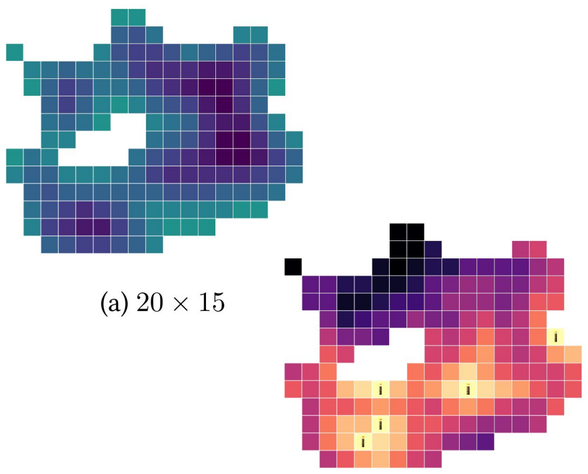
# Quantum Optimization for Solving NP-Hard Problems

Arnav Dayal, Miles Moosavi, Sohan Kosuru, and Raghava Kalidindi

# INTRODUCTION

### What are NP-Hard Problems?

NP-Hard Problems are problems for which no known algorithm can guarantee a solution in deterministic polynomial time. In contrast, problems in P can be solved in polynomial time (e.g., O(n), O($n^2$), etc.), meaning their solutions are predictable and efficient as the input size grows.



(a) 20 × 15



(d) same as above, with torches

### What is Our Problem?

The optimization problem we explored came from a paper that aimed to compute the minimal number of light sources (torches) needed to light up 2D area (represented in 3D via a heightmap in the original paper). Our aim is to generalize the problem to minimize the number of node placements required to "light up" any 2D area; a top-down view over any map.

### How Are We Solving It?

Our researched paper utilized an ADMM algorithm, QUBO, and quantum annealing to achieve its results. Using its findings, we implemented a similar approach that involves simulated annealing to optimize the placements of new access points over a university campus.

# METHODS

### What is Alternating Direction Method of Multipliers (ADMM)?

ADMM is a modified version of the Augmented Lagrangian method, and is a process to solve constrained optimization problems, where given a function with dual variables, the process uses partial updates with a penalty function to find the minimum of a variable.

### What is QUBO?

A Quantum Unconstrained Binary Optimization describes an approach for solving NP-Hard problems; an approach that can implemented via ADMM. It has a wide variety of applications in the field of quantum computing. Essentially, QUBO splits problems into binary and linear sets of problems, and then attempts to find the minima to these problems. Since the minima of these problems are the same as the local minima to all of the problems, minimizing each problem and finding the smallest of these solutions is equivalent to solving a much harder polynomial for it's minima. These minima are analogous to the most optimal solution for the problem.

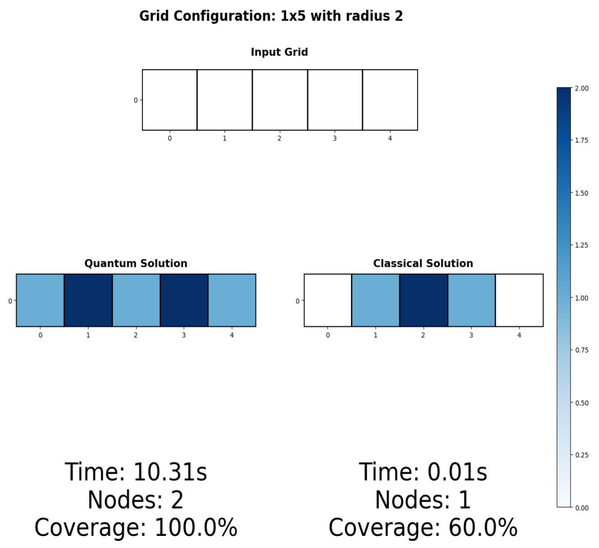
### How Can this Be Implemented as An Algorithm?

We can represent our wifi placement situation, or any 2D mapped area, as a polynomial. Splitting the polynomial into smaller quadratic and linear terms, and then using an alternating direction method we can find the minima of each quadratic or linear term. Then, we can find the smallest minima found, which will give us an answer to the problem.

# RESULTS

### Example Solution:

An example solution is shown here, with a 5 by one grid. The range of the "torch" is 2 pixels wide.




As seen on the left, while the quantum solution does take a much longer time by a factor of a thousand, it does have a better coverage than the classical solution does. Note that the quantum solution is run via a simulator and not on a quantum computer.

### Is This The Solution Optimal?

For a 5x1 with a 2 radius range, the solution given by the quantum computer is the best solution we observed. The classical solution, however, does not adequately cover the whole input grid.

### What Can We Deduce from This?

With regards to coverage, the quantum solution always yielded more consistent results compared to the classical solution. For linear runs, while the quantum version performed better, it always had a much longer runtime compared to the classical solution.

| Grid Size by Radius | Quantum Nodes | Quantum Coverage | Classical Nodes | Classical Coverage | Time ratio (quantum/classical) |
|---|---|---|---|---|---|
| 1x3 by 2 | 1 | 100% | 1 | 66.6% | 114:1 |
| 2x2 by 2 | 1 | 100% | 1 | 100% | 209:1 |
| 2x3 by 2 | 1 | 100% | 1 | 100% | 7390:1 |

### Time and Qubit Cost

Currently, this algorithm takes a lot of time compared to classical computing. More concerningly, it takes an exponentially large amount of qubits (taking 2^n qubits for n cells). While normal quantum computers range from about 30 to 100 qubits, quantum annealers (which this process was designed for in the original paper) have thousands of qubits.

### Size Limitations

Due to the above issues, the amount of cells we can actually run is quite limited. This limits the sizes we can test for ideal placements.

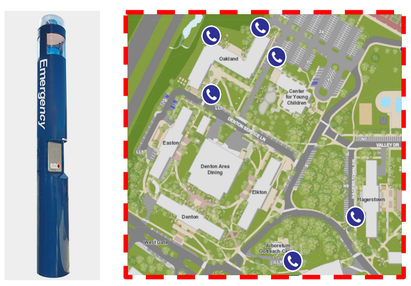
# CONCLUSION

### Significance of Optimization:

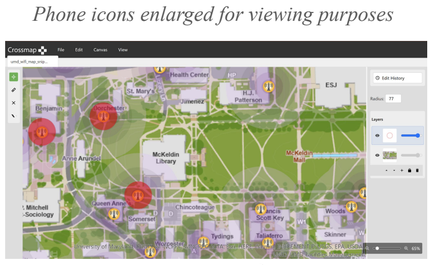
Optimizing placements, such as positioning wells, stores, or Wi-Fi access points, is common in many fields. Our approach faces challenges due to limited quantum computing access, but advancements in qubit capacity should overcome these obstacles in the future.

### Practical Applications and Future Work:

A web application, while currently in development, would allow users to select an area to optimize based on its current available resources. It would allow for more accessible interface towards achieving the quantum results as seen in this paper.




*Phone icons enlarged for viewing purposes*



crossmap.xyz demo

# REFERENCES

**Efficient Light Source Placement using Quantum Computing:** https://arxiv.org/abs/2312.01156

**Quantum Bridge Analytics I: A Tutorial on Formulating and Using QUBO Models:** https://arxiv.org/ftp/arxiv/papers/1811/1811.11538.pdf

### Additional Resources:

https://qiskit-community.github.io/qiskit-optimization/

https://web.stanford.edu/class/ee364b/lectures/admm_slides.pdf

https://www.youtube.com/watch?v=h8YKBVmRW-E

https://maps.umd.edu/map/index.html?&x=-76.94220704760498&y=38.99188853363314&zoom=15&basemap=detailed&layers=pert

### Our Github:

https://github.com/MilesMoosavi/crossmap