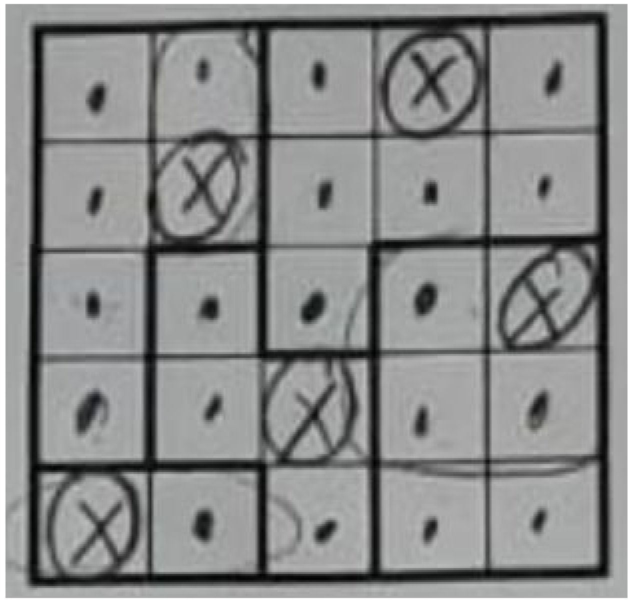
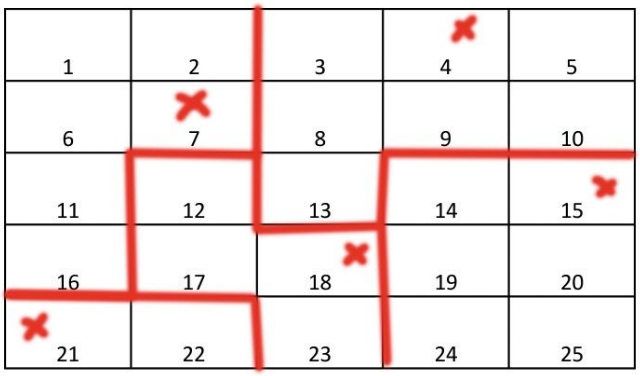
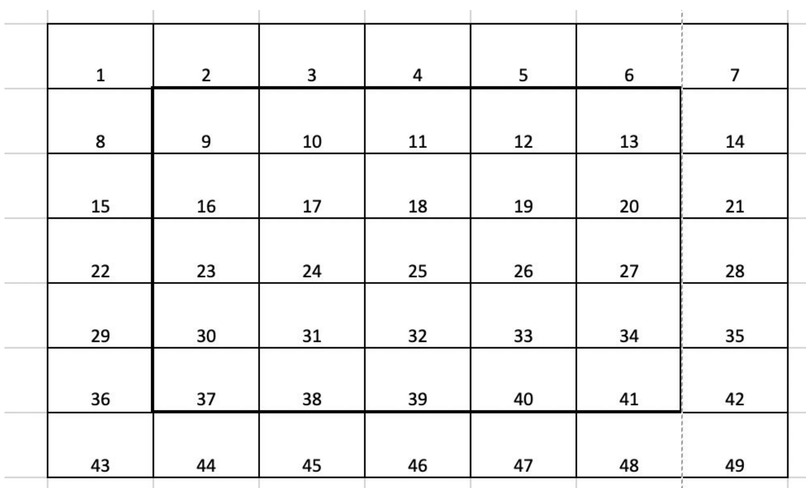


Figure 1: Solved 5x5 Star Battle game.

The problem that our group took on was to implement the game titled “Star Battle” . It is a game that has rules similar to Sudoku. The game is typically played on an 8x8 grid that is partitioned into various sections. The goal is to place stars on the grid so that there is exactly one star per row, column, and partition. No stars may be adjacent to each other, even diagonally. In order to simplify this problem, a 5x5 matrix was considered. A solved puzzle is shown in Figure 1. The goal was to create a QUBO that was minimized when the stars are in the correct configuration.

On the 5x5 grid, there are a total of 25 possible locations to place a star. However, only a total of five stars may be placed. Therefore, if a position is represented as a matrix as a variable x_i, a position with a star can be represented by a value of 1 while one without a star would be zero. Therefore, the total summation of the grid must be minimized to equal five to ensure that the star requirement has been reached. There are three constraints that have to be considered. SInce each row, column, and partition must contain only one star, then the sum of the five terms in each of these must not exceed five to give a proper solution. This objective and these constraints are what laid the foundation for the QUBO. The result that came back with the lowest energy would be represented as a valid solution.





We expanded the 5 by 5 structure to 7 by 7 so that we could more easily to do the calculation

We create 15 equations to set each row, each column, each set to only have one star.

For example,

For row 1 we set $x_9 + x_{10} + x_{11} + x_{12} + x_{13} = 1$

For column 1 we set $x_9 + x_{16} + x_{23} + x_{30} + x_{37} = 1$

For set 1 we set $x_9 + x_{10} + x_{16} + x_{17} + x_{23} + x_{30} = 1$

And we set 25 constraints to make sure there are no two stars nearby.

For example, for block 1 we set $x_1 + x_2 + x_3 + x_8 + x_9 + x_{10} + x_{15} + x_{16} + x_{17} \leq 1$

Which represent the block centered at 9. And we will do that for each index in the center 5 by 5 part.

QUBO:

Objective: Minimize the sum of all of the elements in the game board

$$\min\left(\sum_{i=1}^{25} x_i\right) \qquad x_i = \begin{cases} 1, & \text{if star exists} \\ 0, & \text{otherwise} \end{cases}$$

Constraints:

1) Only 1 star per row, column, and region.

$$\sum_{k}^{15}\left(25 - 10\sum_{i}^{|S_k|} x_i + \sum_{j \neq i}^{|S_k|} x_i x_j\right)$$

2) No star can be adjacent to any other star.

$$\sum_{k}^{40}\left(\sum_{i}^{9}\sum_{j<i} x_i x_j\right)$$

$$\text{Qubo: } \sum_{i=1}^{25} x_i + P\left[\sum_{k}^{15}\left(25 - 10\sum_{i}^{|S_k|} x_i + \sum_{j \neq i}^{|S_k|} x_i x_j\right) + \sum_{k}^{40}\left(\sum_{i}^{9}\sum_{j<i} x_i x_j\right)\right]$$

The primary barrier was the implementation of the QUBO itself. Even with the size reduction of the grid, the length caused the QUBO to include several summation terms that complicated its implementation into Python. For future directions, we intend to fully implement the QUBO and run it to see if it would return valid solutions as the lowest energy given.