Dreams AI Task Mathematical Proof

Jonathan Wang

February 22, 2022

1 Elimination Game Proof

We can first represent the board states with a binary matrix of mole states where $s \in \{0, 1\}$. 1 represents an exposed mole and 0 represents a hiding mole.

Therefore, an example matrix could be as follows:

$$\text{mole state} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \tag{1}$$

Hitting a mole is therefore equivalent to binary addition with what we will define as an adjacency matrix, for example, adjacency matrices for the squares (0,0) and (1,2) look as follows:

$$\mathbf{ADJ}_{(0,0)} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{ADJ}_{(1,2)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{2}$$

These adjacency matrices can form the basis matrices with which we can create the initial mole state.

For the sake of computational efficiency, we can convert this visualizable matrix representation into vector notation where each vector has length n^2 where n is the size of the square matrix.

$$\vec{adj}_{(1,2)} = [0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 0] \tag{3}$$

We also have a constraint that matrix addition can only be performed when the square the adjacency matrix targets is in the state 1 i.e. we can only hit a mole when it is in the 'up' state.

Generalizing further we can obtain a new set of adjacency matrices, e.g. if the rules state that the diagonal neighbours are flipped instead, we can have a new set of basis matrices:

$$\mathbf{ADJ}_{(1,2)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \tag{4}$$

An initial solution to this problem could be finding a combination of basis vectors through gaussian elimination that can represent the initial state. The order of operation must be such that at each step, the (i,j) adjacency matrix must have a square which is in state 1.

Hence, the solution would be in the form:

$$\text{Initial Mole State} + \sum_{i,j} x_{i,j}\mathbf{ADJ}_{(i,j)} = 0 \tag{5}$$

Casting this into a linear algebra problem (easy since it is a binary operation) which is computationally efficient we get:

$$\mathbf{A}x = b \tag{6}$$

where $\mathbf{A}$ is the adjacency matrix, x is the number of times we need to click on each tile and b is the initial state. Since this is a binary problem, x will always be $\in \{0, 1\}$. All computations are effectively XOR logic computations done over GF(2).

A solution to this would be through gaussian elimination using the 3 elementary row operations:

1. Swap two rows
2. Multiply a row by a scalar (not relevant for binary)
3. Add one row to another

Doing this for our initial state (1) yields the vector:

$$x = [0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1] \tag{7}$$

Therefore, we need to press mole 6, 14 and 16 in order to eliminate them all. Now, we must reconsider the constraint imposed that each action must only occur on the tiles that are in state 1. We must find the shortest sequence that results in a success. A simple approach would be to model all possible permutations and combinations of moves as nodes from a path from start to finish, terminating the path if it results in an illegal move.

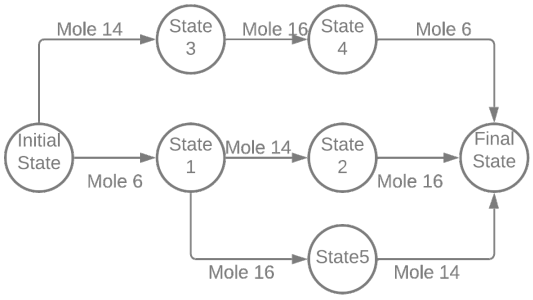

Figure 1: Example graph with nodes

Finally, the shortest path solution to this graph can be found through traditional algorithms such as dijkstra's algorithm assuming all nodes have path length 1.