Shedding Some Light on the Light Up Puzzle with Artificial Intelligence

James Browning
Samuel Ginn College of Engineering
Auburn University
Auburn, AL 36849
jlb0181@auburn.edu

Roberto Perera
Samuel Ginn College of Engineering
Auburn University
Auburn, AL 36849
rzp0063@auburn.edu

Libo Sun
Samuel Ginn College of Engineering
Auburn University
Auburn, AL 36849
lzs0101@auburn.edu

1 Proposal

For our final project, we will attempt to develop an artificial intelligence capable of solvings the light up logic puzzle, also known as "Akari." The game works like this:

1. Each puzzle begins with a rectangular grid of black and white cells.

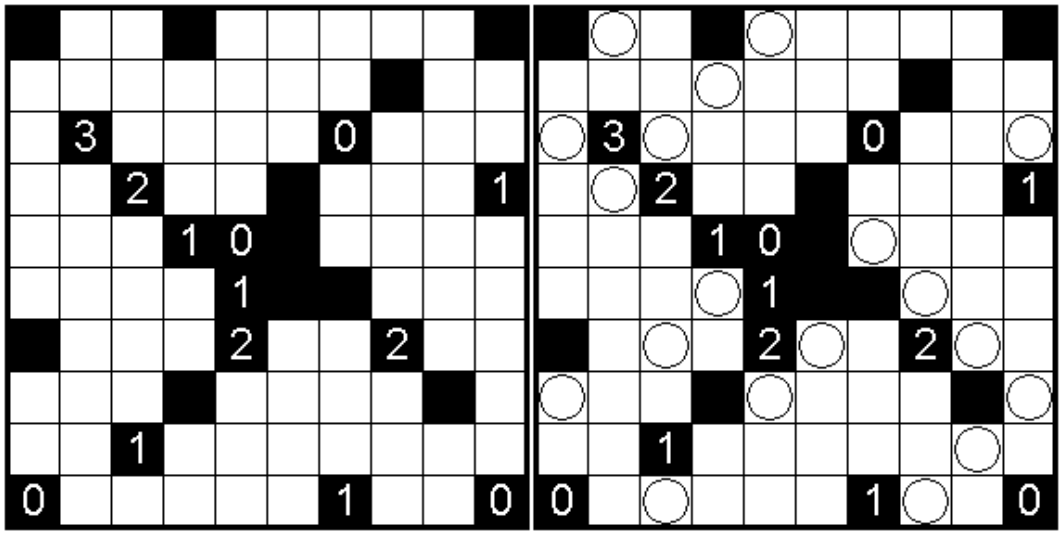
2. Players must place light bulbs in white cells, in an arrangement such that:

   (a) Every white square is illuminated (light bulbs send rays of light up, right, down, and left, unless blocked by a black square).

   (b) No two light bulbs shine on each other.

3. Players must pay attention to the numbers inside some black squares, which impose a limit on how many light bulbs can be placed in adjacent white squares.

Below is an example board (left) and its solution (right), taken from Wikipedia. Developing simulators for the board is simple enough, so we will be able to focus on solving the problem, likely through an informed search algorithm like hill-climbing plus simulated annealing.



Initial states: The boards will be generated at random using user-defined inputs, such as board dimensions and number of black cells.

Approach: First, lightbulbs will be placed at random initial positions, based on the given limit of adjacent bulbs for each black square. The board may contain some initial violations (such as two bulbs shining on each other).

Output: The system will attempt to produce an optimal solution, based on the performance measure, using several different search algorithms. Allowed moves will include placing a bulb on an empty cell and removing a bulb from a filled cell.

Algorithms: First, we will implement informed search algorithms — including various forms of hill climbing, as well as simulated annealing — to determine what moves to make. Second, after the simulated annealing method has been implemented successfully, a large dataset of randomly generated initial states, as well as the moves needed to solve them, will be saved by running the simulated annealing for approximately one week. Ideally, this will generate enough training data to make training a deep neural network possible, with the goal of solving the puzzle given any initial state. Our definition of performance will include how many white squares a move lights up (increasing performance), as well as how many violations it might cause (decreasing performance).

Baselines: Baselines will include various methods of hill climbing, simulated annealing, and deep neural networks. To compare their performance, we plan to analyze factors such as the total average time it takes for each method to solve a series of puzzles, the average number of moves made, and the average performance cost. It will be extremely important to explore various tuning parameters for each algorithm, such as temperature and probability for simulated annealing, and number of hidden layers, number of neurons per hidden layer, learning rate, and activation functions (e.g. cross entropy, sigmoid) for deep neural networks. Additionally, other game-specific parameters must be tuned, such as the number of neighbors which are generated for selection, conditions of search termination, and choice of hill climbing variant.

According to research by Brandon McPhail [1], the light up puzzle is NP-complete, though it is not discussed nearly as much in artificial intelligence cycles as problems like eight queens — making it a perfect problem to solve for a project like this.

References

[1] McPhail, B. (2005, February 28). Light Up is NP-Complete. Retrieved October 10, 2020, from http://mountainvistasoft.com/docs/lightup-is-np-complete.pdf