Introduction to Artificial Intelligence: Local Search

Nathan Morgenstern, Seo Bo Shim

September 21, 2017

Contents

Introduction to the Graphical User Interface(GUI)

When the Graphical User Interface starts up the user is able to select the type of puzzle evaluation through a drop down menu. The given options include: Basic Puzzle Evaluation, User Generated Puzzle Evaluation, Basic Hill Climbing, Hill Climbing with Random Restarts, Hill Climbing with Random Walk, Simulated Annealing, and Population Based Approach.

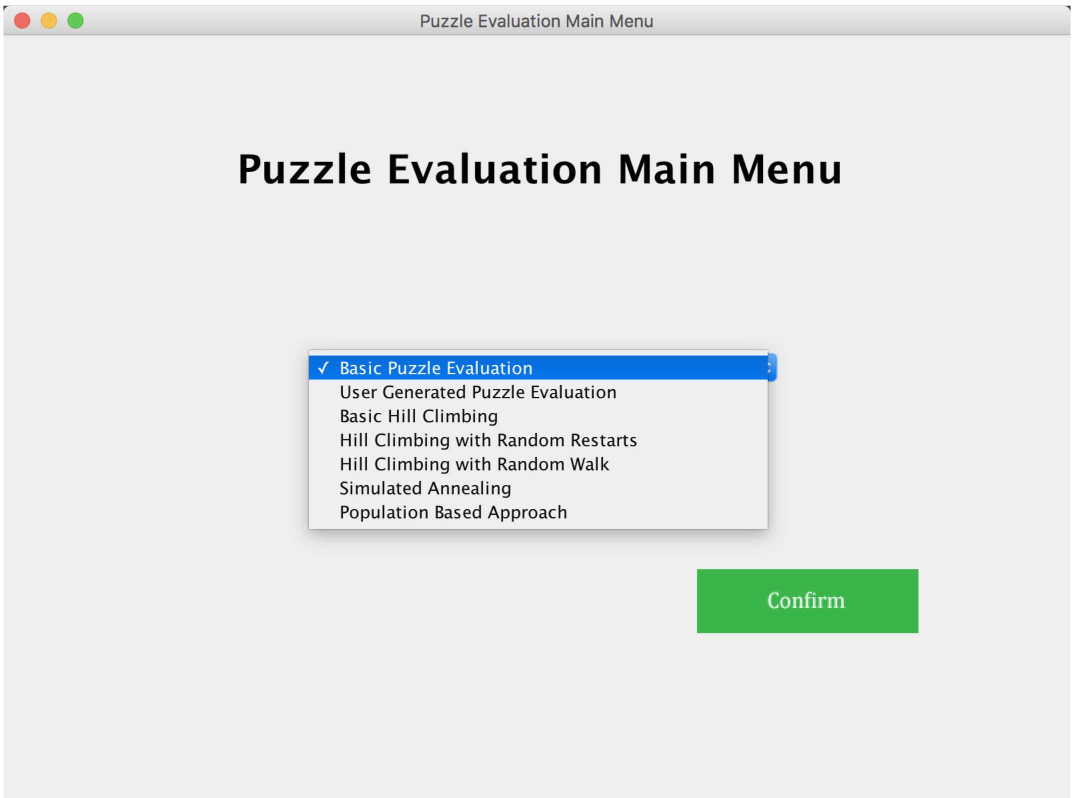

Figure 1: Main Menu of GUI

Each of the options then have their own corresponding window which is comprised of four main tabs: Puzzle Initialization, Puzzle, Puzzle Moves, and Puzzle Data. The Puzzle Initialization tab is slightly different for each option in regards to the type of input received. The tabs Puzzle and Puzzle moves provide the user with a graphical representation of the generated puzzle as well as a graphical representation of the number of moves that it takes to get to each cell respectively.

Basic Puzzle Evaluation

The Basic Puzzle Evaluation creates a randomly generated puzzle that contains only legal moves. The Basic Puzzle Evaluation window asks the user to provide a size n for the puzzle, where n can take on the values 5, 7, 9, and 11. They are then able to generate the puzzle by clicking the generate button. Each time generate is selected a random puzzle of the selected size is generated.

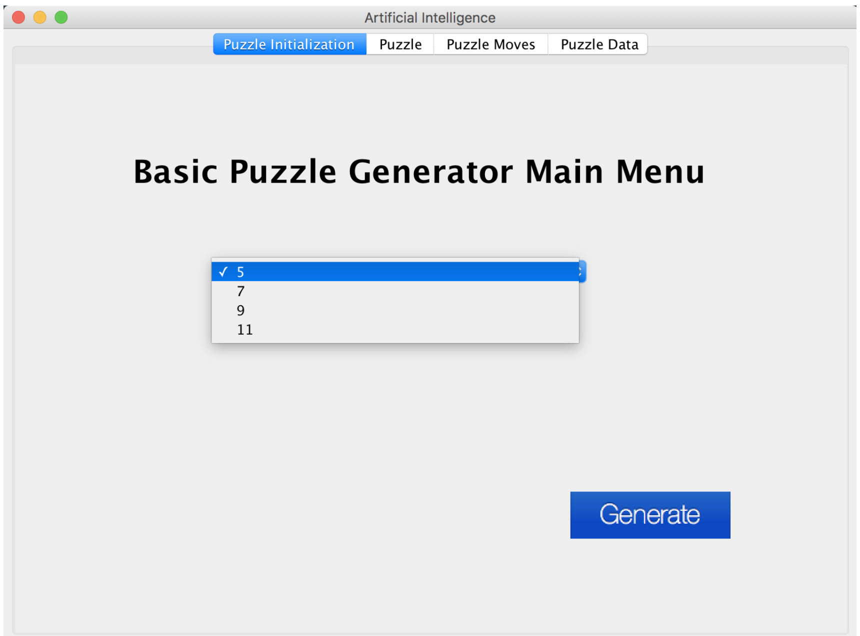

Figure 2: Puzzle Initialization of Basic Puzzle Evaluation Option

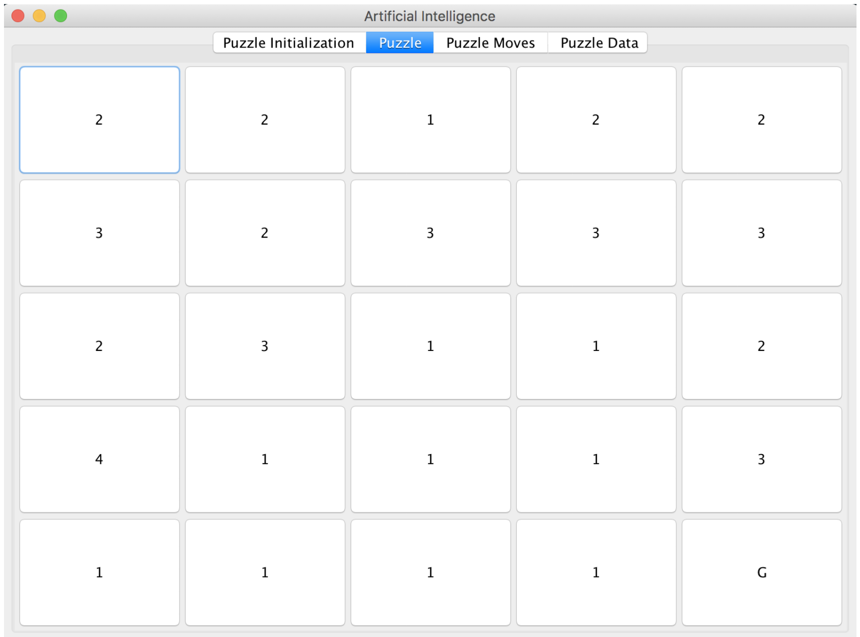

Figure 3: Puzzle Tab of Basic Evaluation

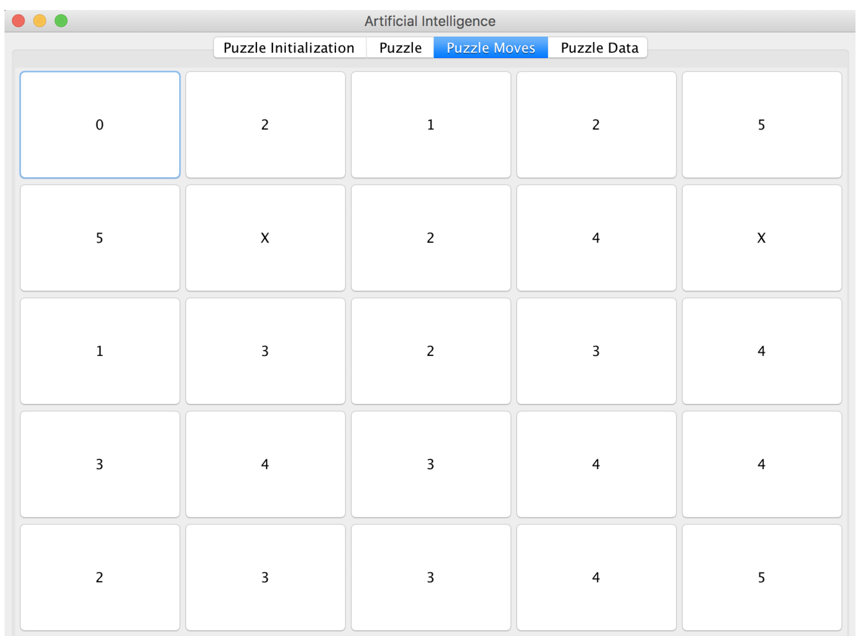

Figure 4: Puzzle Moves Tab of Basic Evaluation

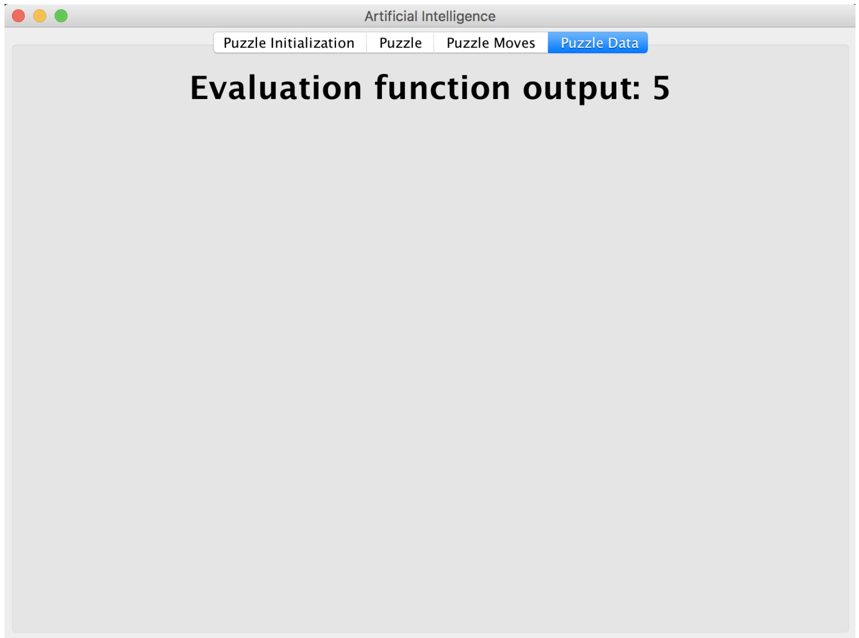

Figure 5: Puzzle Data Tab of Basic Evaluation

User Generated Puzzle Evaluation

The User Puzzle Evaluation includes a text field where the user can input the filepath to a pregenerated puzzle for the program to solve. It will then visualize the puzzle, puzzle moves, and result of the evaluation function in the corresponding windows.

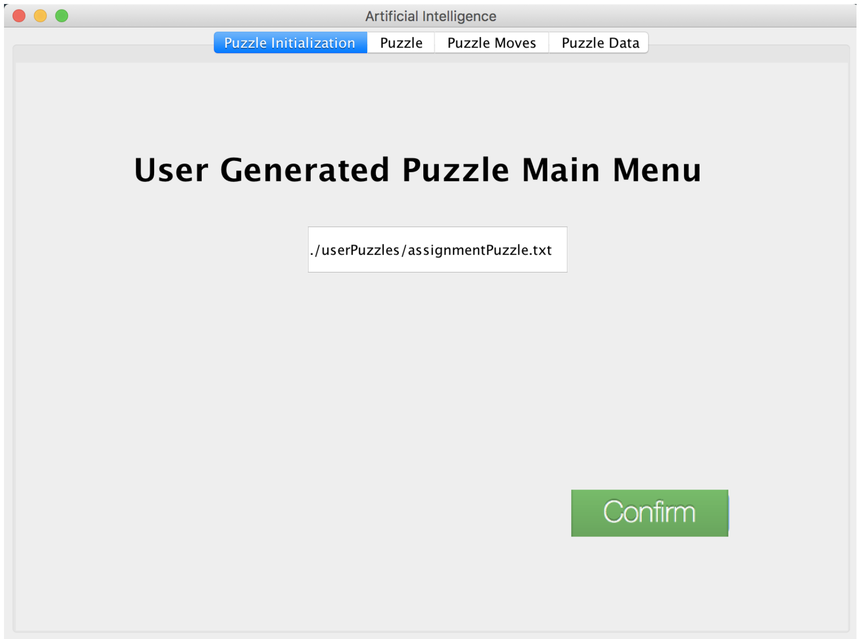

Figure 6: User Generated Puzzle Initialization

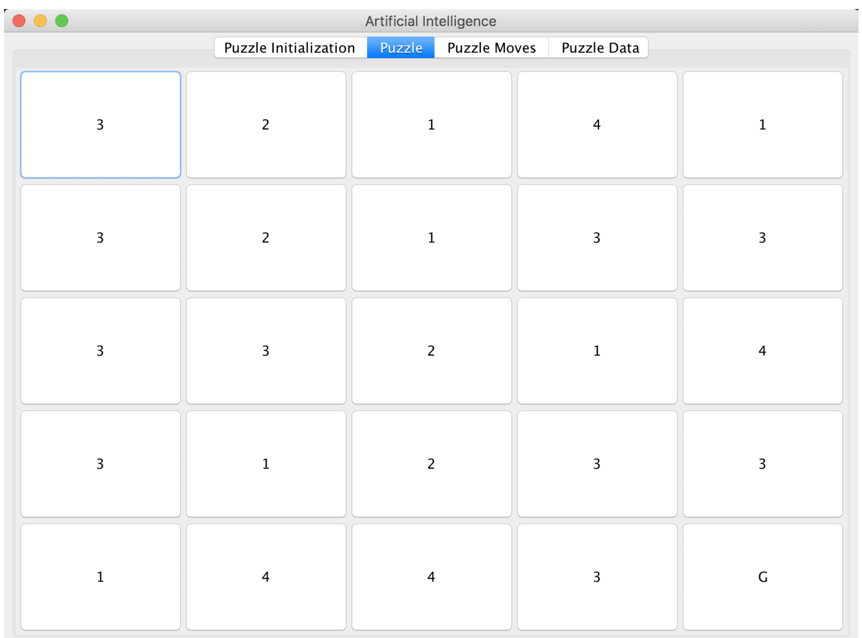

Figure 7: User Generated Puzzle

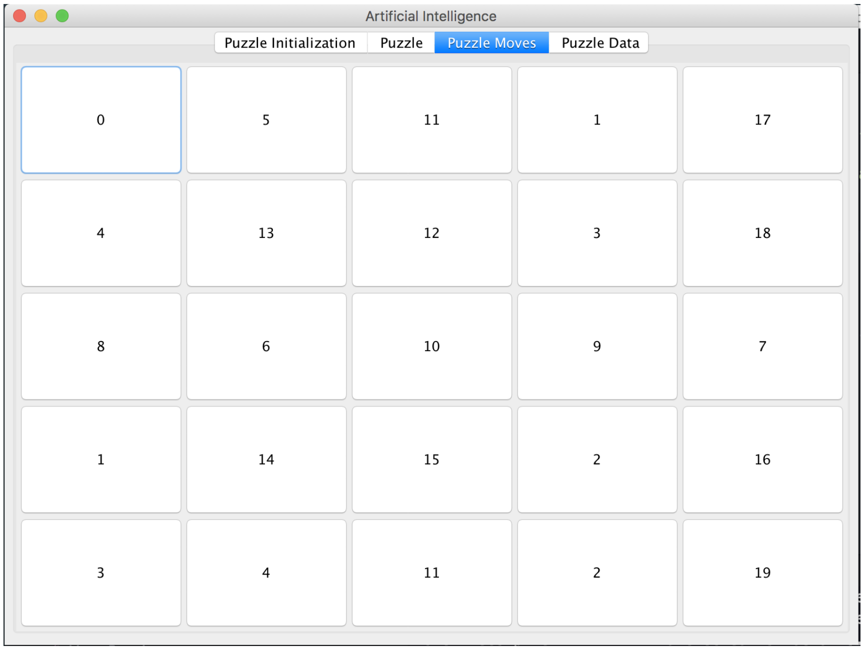

Figure 8: User Generated Puzzle Moves

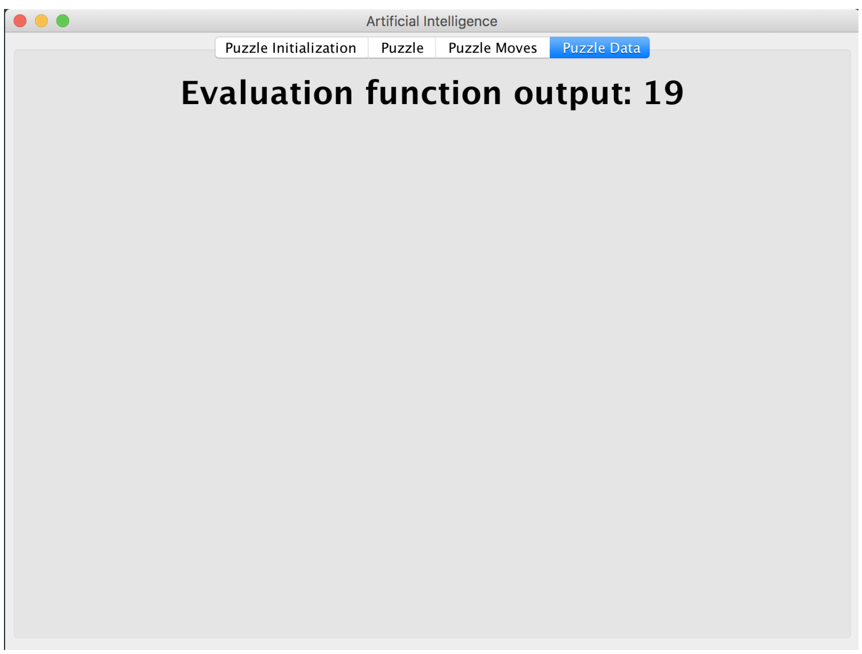

Figure 9: User Generated Puzzle Evaluation

Basic Hill Climbing

The Basic Hill Climbing option similiarly to the Basic Puzzle Evaluation option includes a box that allows the user to choose a puzzle size n, as well as an option that lets the user choose the desired number of iterations for the hill climbing algorithm. The puzzle and puzzle moves tabs remain unchanged. The puzzle data tab still includes the evaluation output and additionally gives the estimated amount of time the algorithm took to run in miliseconds.

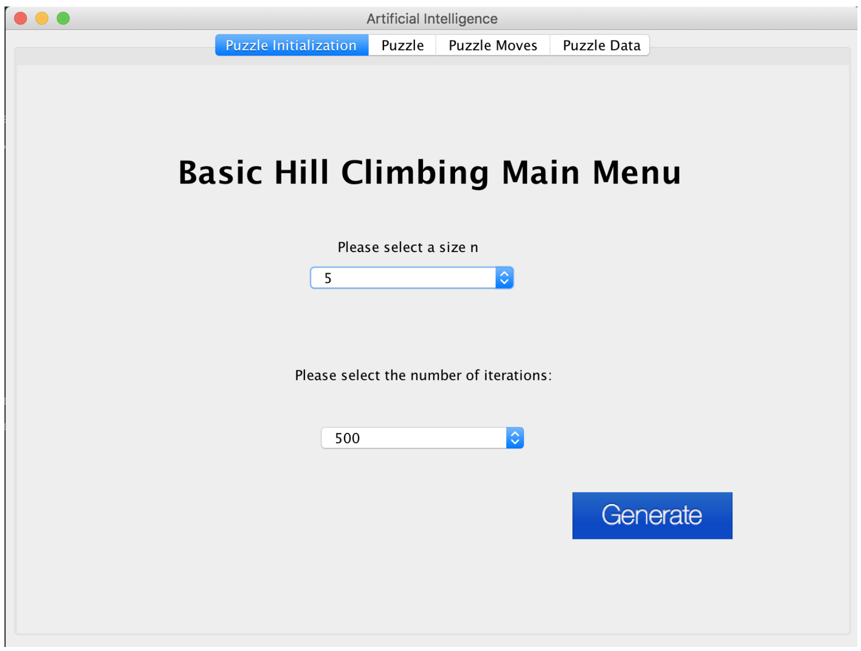

Figure 10: Basic Hill Climbing Puzzle Initialization

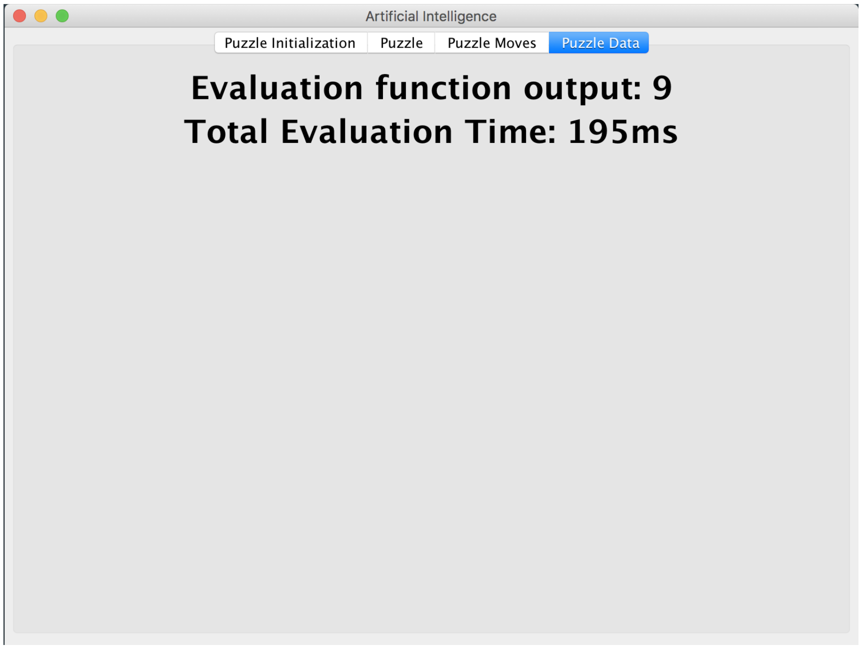

Figure 11: Basic Hill Climbing Puzzle Data Tab

Hill Climbing with Random Restarts

Hill Climbing with Random Walk

Simulated Annealing

Population Based Approach

Puzzle Representation

The Graphical User Interface starts up and gives the user

Puzzle Evaluation

Example solvable and unsolvable puzzles for size n = 5, 7, 9, 11 are included below showing the Puzzle, Puzzle Moves, and Puzzle Data tabs for each case. The GUI shows the same type of output for the options Basic Puzzle Evaluation Puzzle and User Generated Puzzle Evaluation. Please see Figure 6 to Figure 9 for an example of a User Generated Puzzle.

Example Puzzle for n = 5

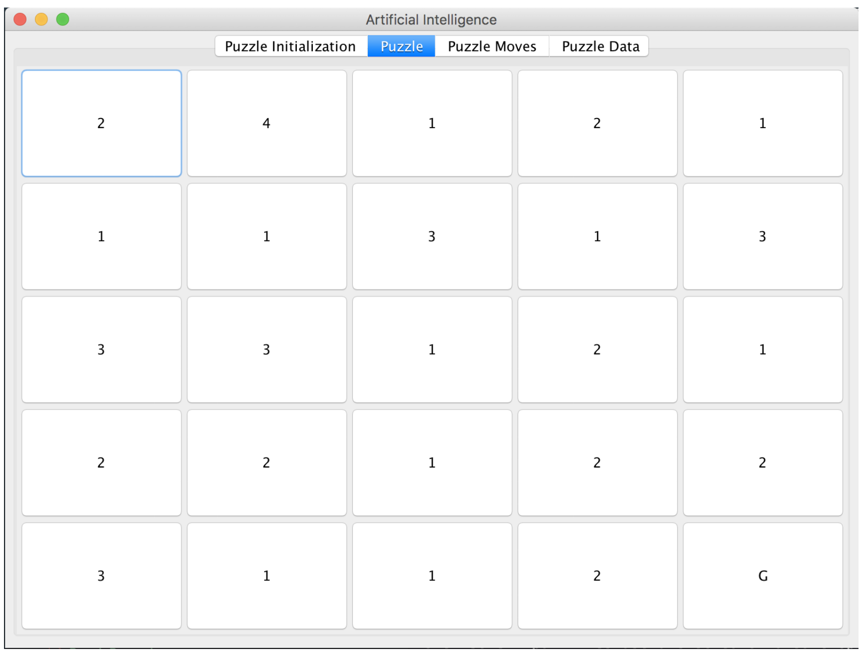

Figure 12: Reachable Goal Puzzle size n = 5

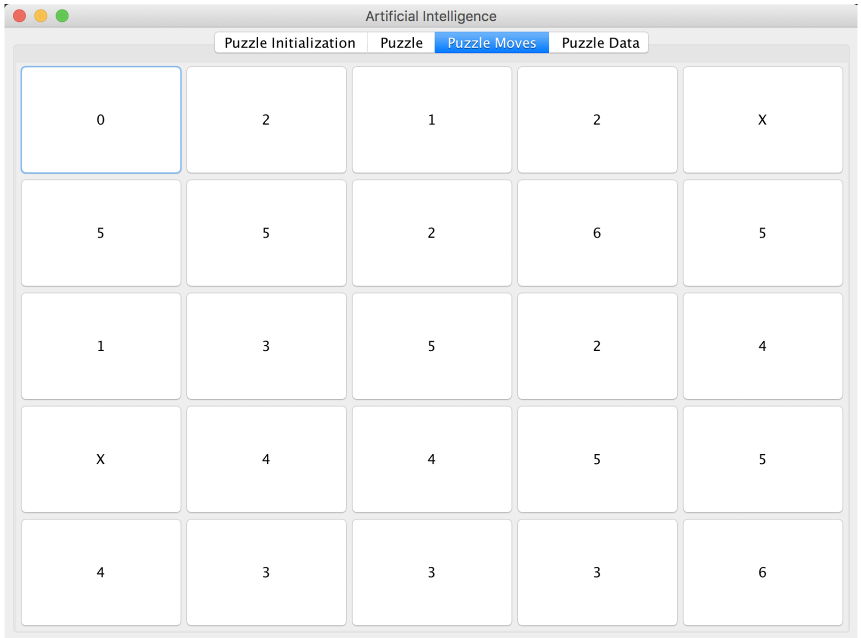

Figure 13: Reachable Goal Puzzle Moves size n = 5

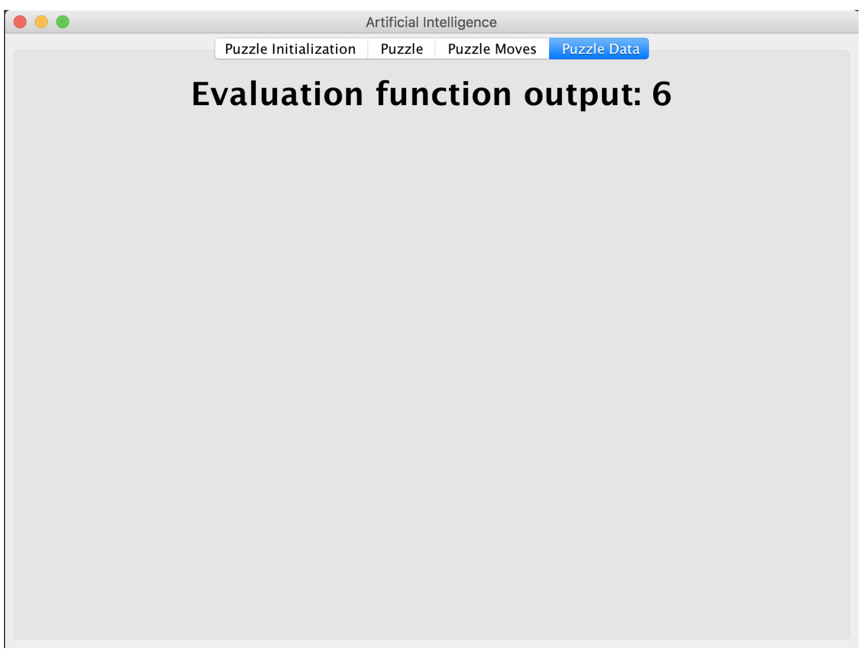

Figure 14: Reachable Goal Puzzle Evaluation size n = 5

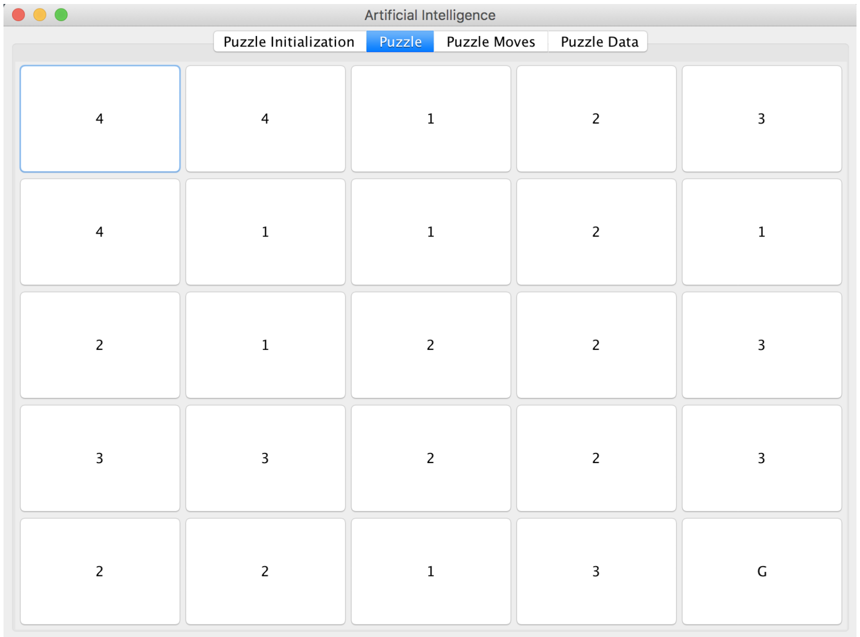

Figure 15: Unreachable Goal Puzzle size n = 5

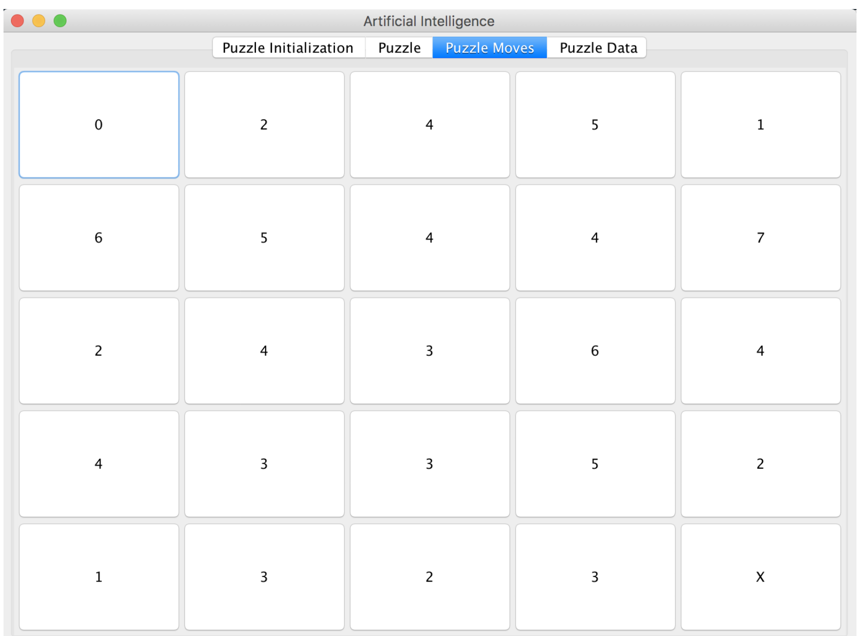

Figure 16: Unreachable Goal Puzzle Moves size n = 5

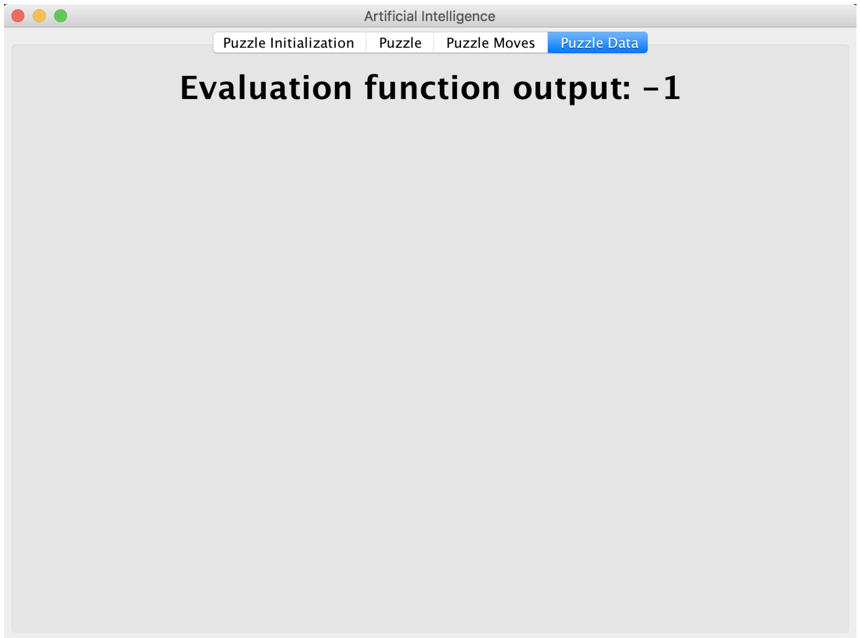

Figure 17: Unreachable Goal Puzzle Evaluation size n = 5

Example Puzzle for n = 7

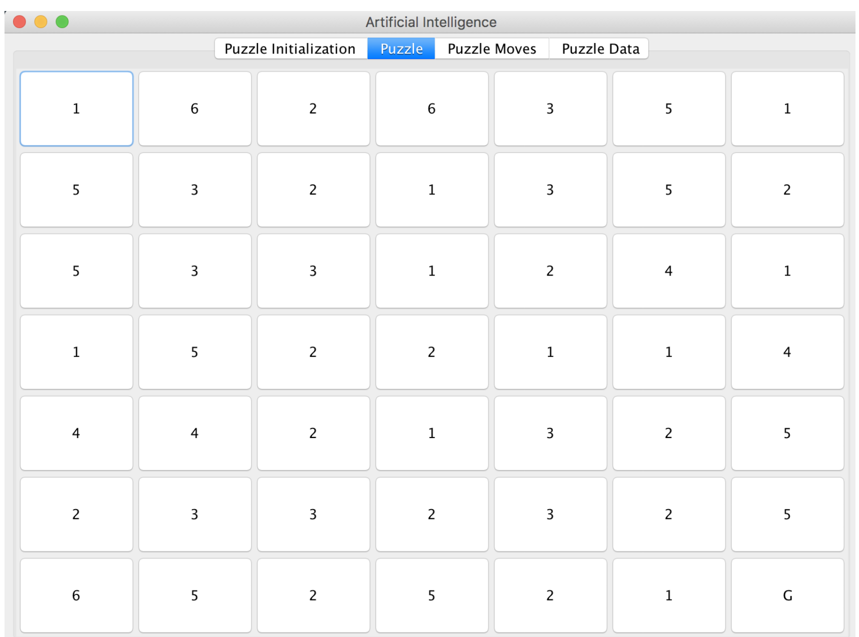

Figure 18: Reachable Goal Puzzle size n = 7

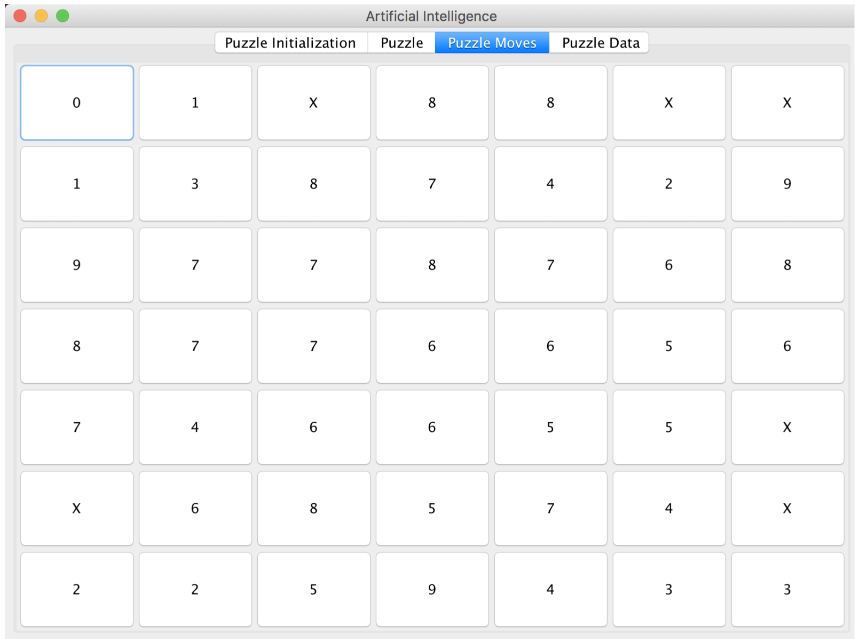

Figure 19: Reachable Goal Puzzle Moves size n = 7

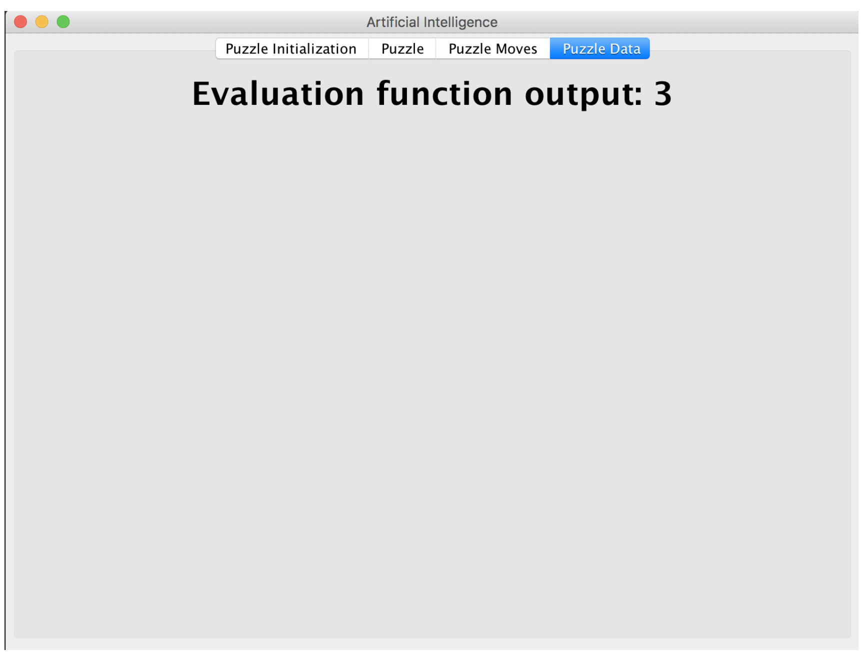

Figure 20: Reachable Goal Puzzle Evaluation size n = 7

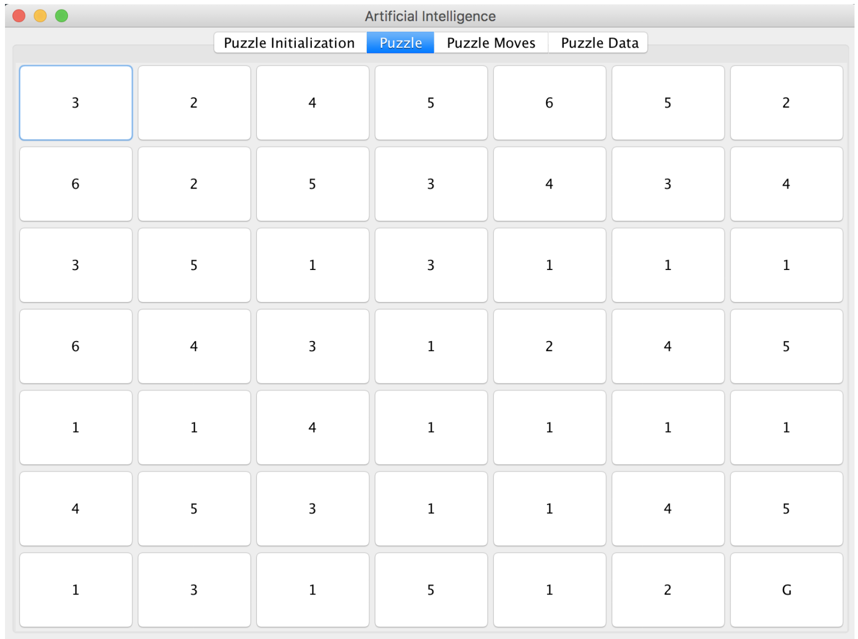

Figure 21: Unreachable Goal Puzzle size n = 7

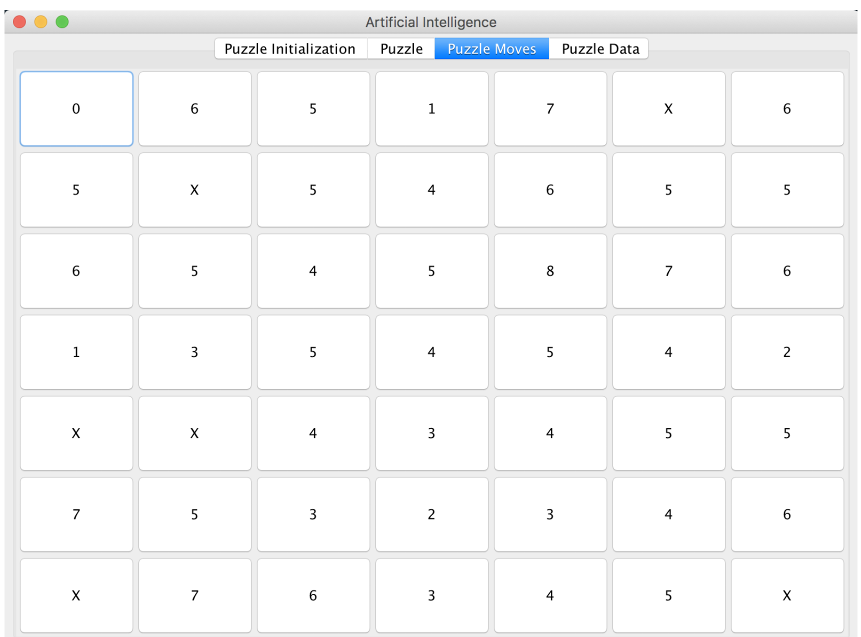

Figure 22: Unreachable Goal Puzzle Moves size n = 7

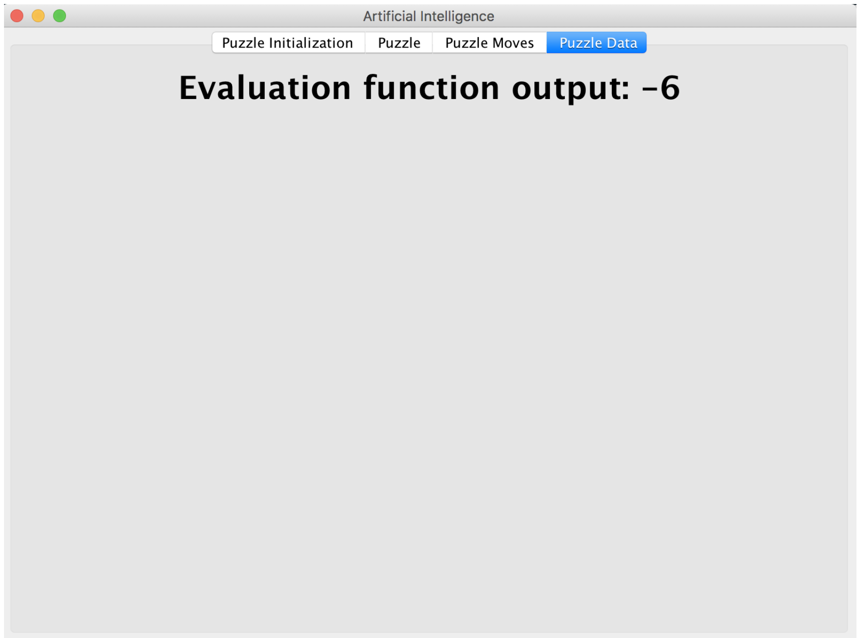

Figure 23: Unreachable Goal Puzzle Evaluation size n = 7

Example Puzzle for n = 9

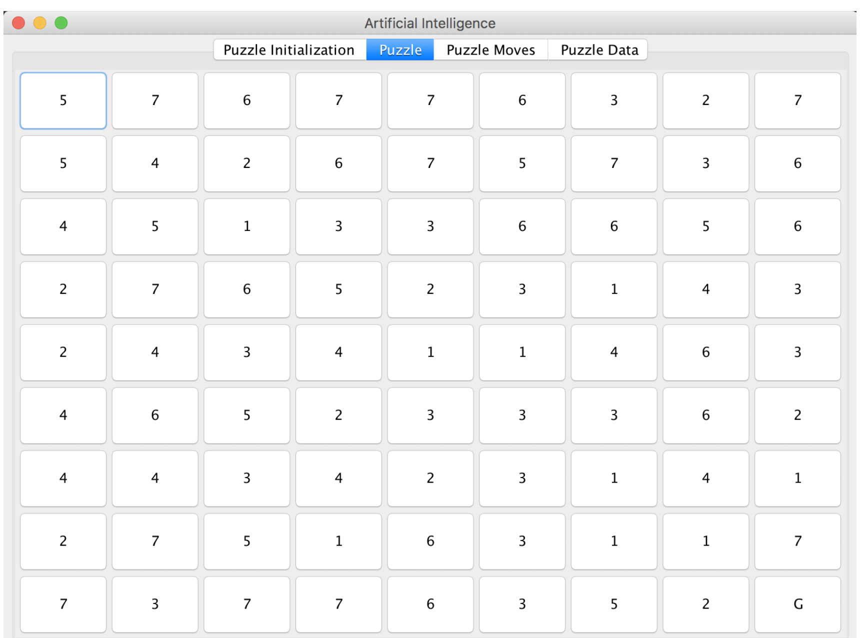

Figure 24: Reachable Goal Puzzle size n = 9

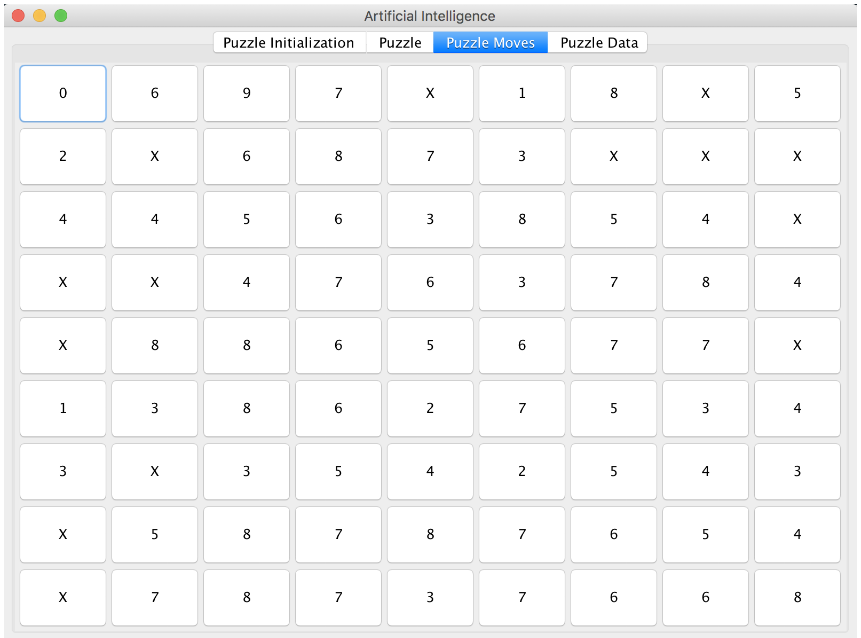

Figure 25: Reachable Goal Puzzle Moves size n = 9

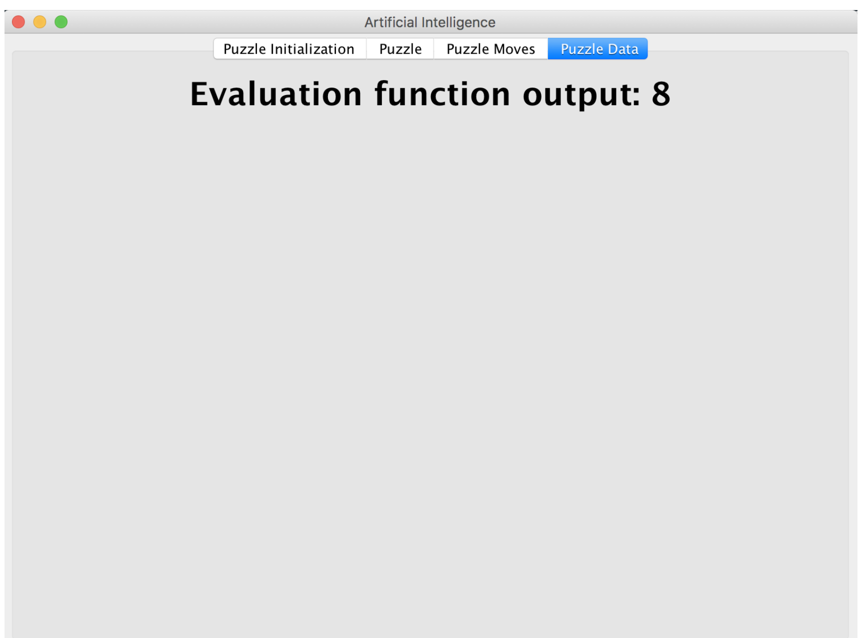

Figure 26: Reachable Goal Puzzle Evaluation size n = 9

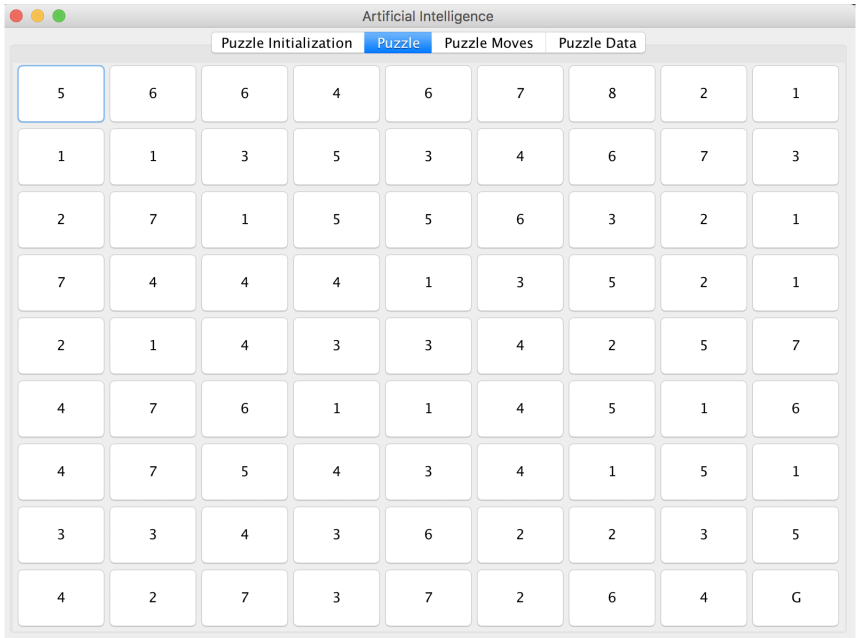

Figure 27: Unreachable Goal Puzzle size n = 9

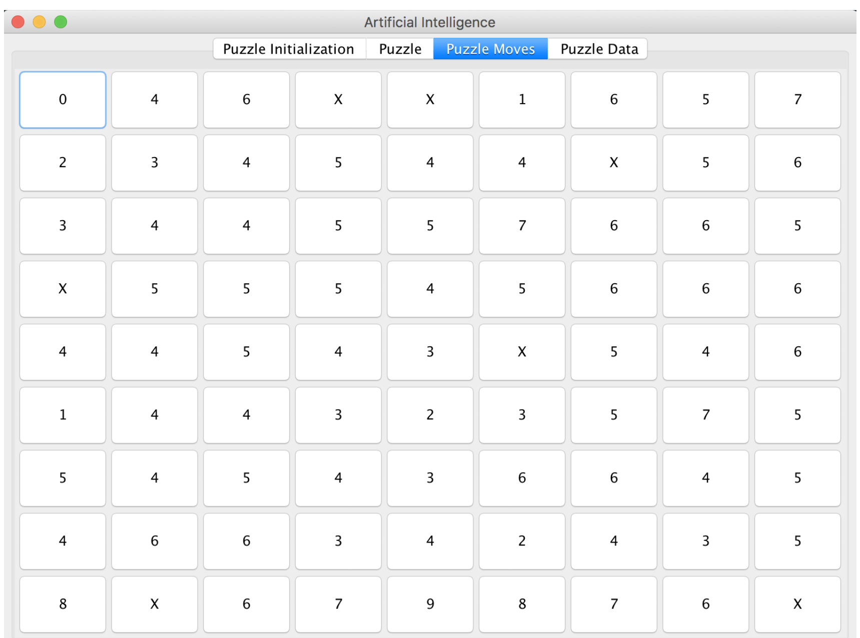

Figure 28: Unreachable Goal Puzzle Moves size n = 9

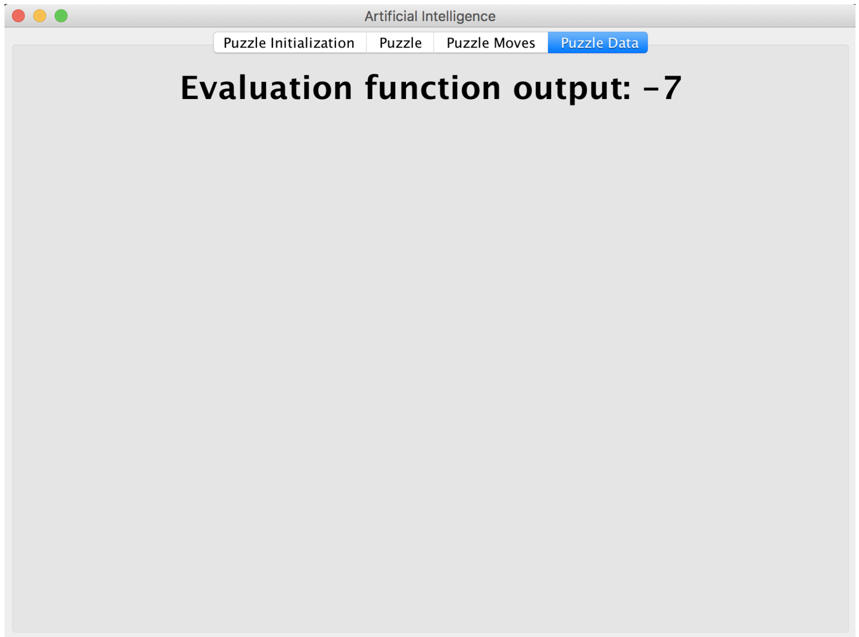

Figure 29: Unreachable Goal Puzzle Evaluation size n = 9

Example Puzzle for n = 11

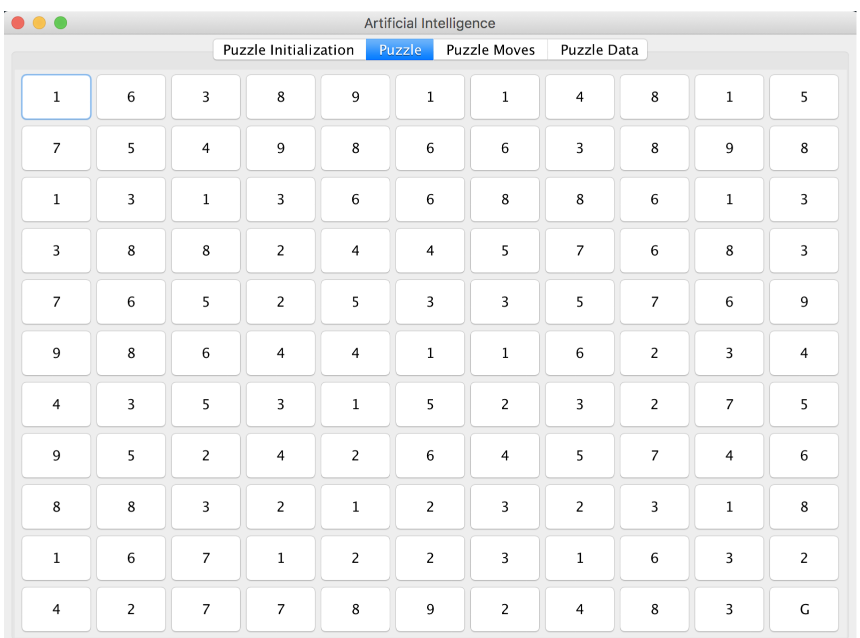

Figure 30: Reachable Goal Puzzle size n = 11

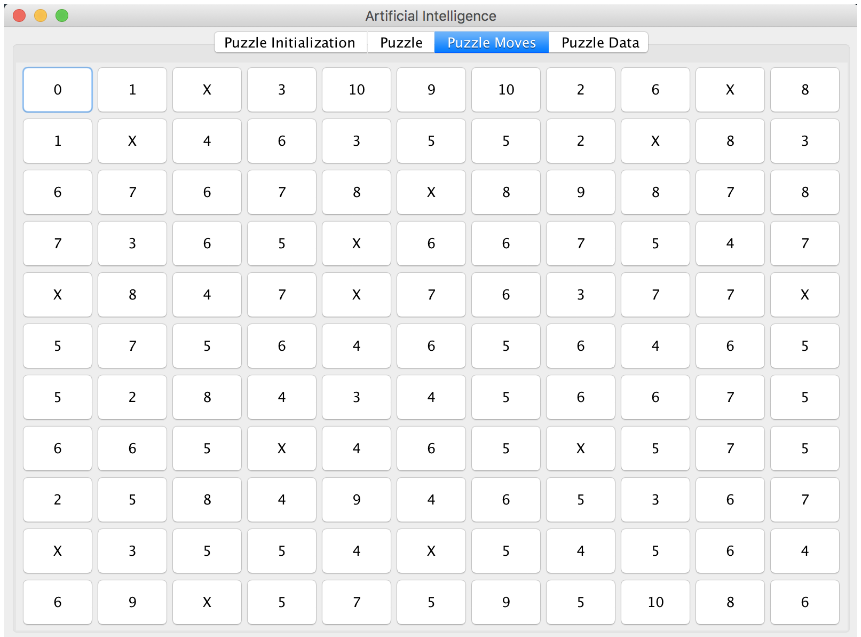

Figure 31: Reachable Goal Puzzle Moves size n = 11

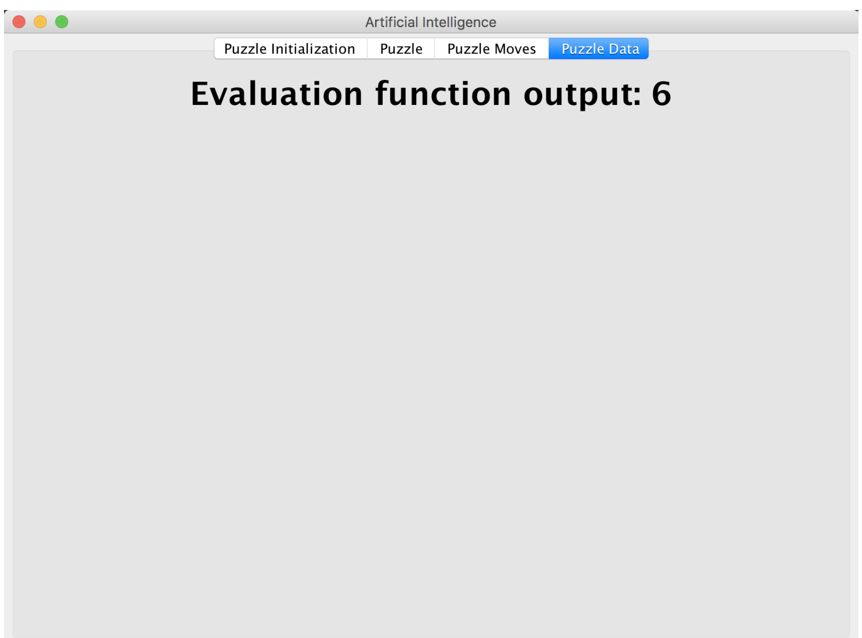

Figure 32: Reachable Goal Puzzle Evaluation size n = 11

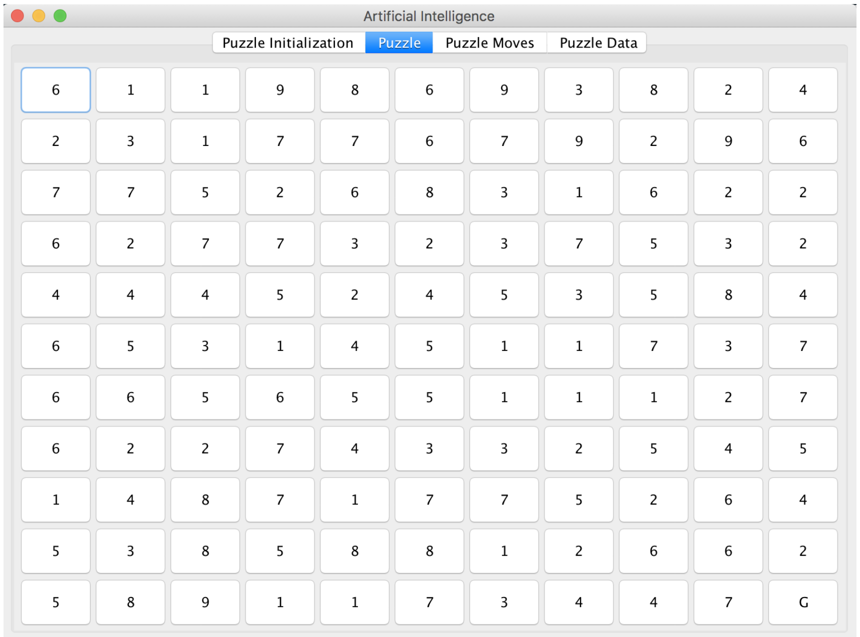

Figure 33: Unreachable Goal Puzzle size n = 11

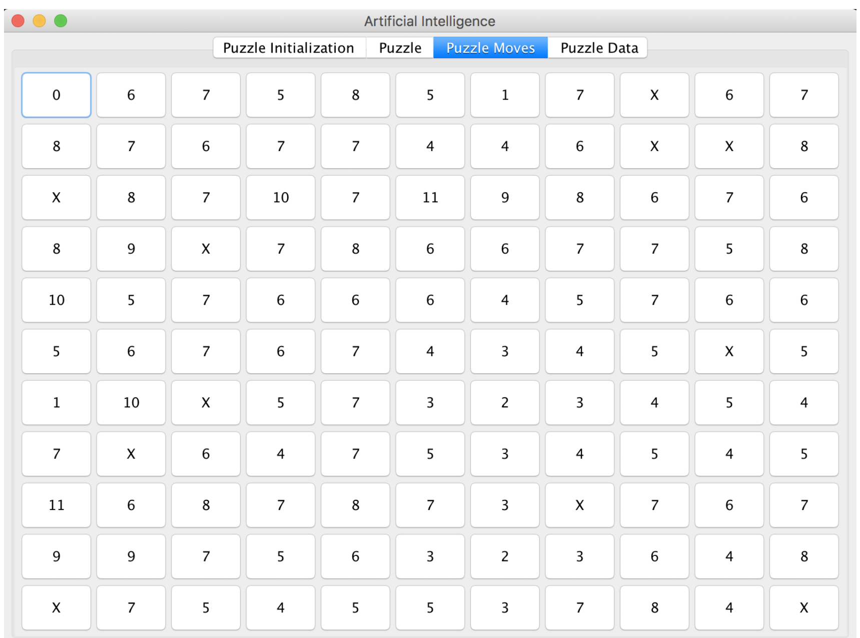

Figure 34: Unreachable Goal Puzzle Moves size n = 11

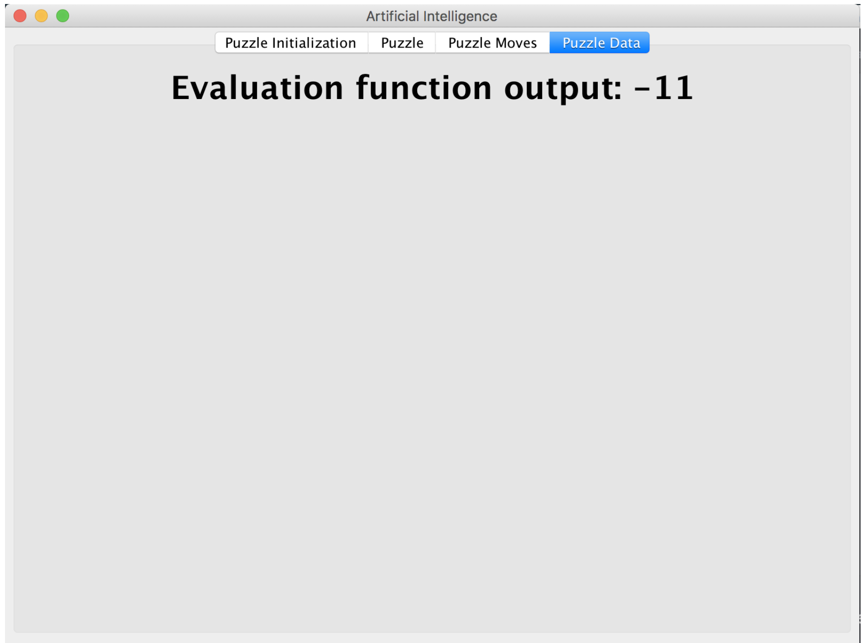

Figure 35: Unreachable Goal Puzzle Evaluation size n = 11

Basic Hill Climbing Approach

The Hill Climbing is able to create puzzles that are on average more difficult then the single random puzzle. One of the downfalls of this algorithm is that it tends to get stuck around extrema. This can be seen from the graphs that generally show the number of goal moves starting low-typical of the random puzzle-and then quickly going up and plateauing. These plateaus are occasionally surpassed although it typically takes many iterations.

Example Puzzle for n = 5

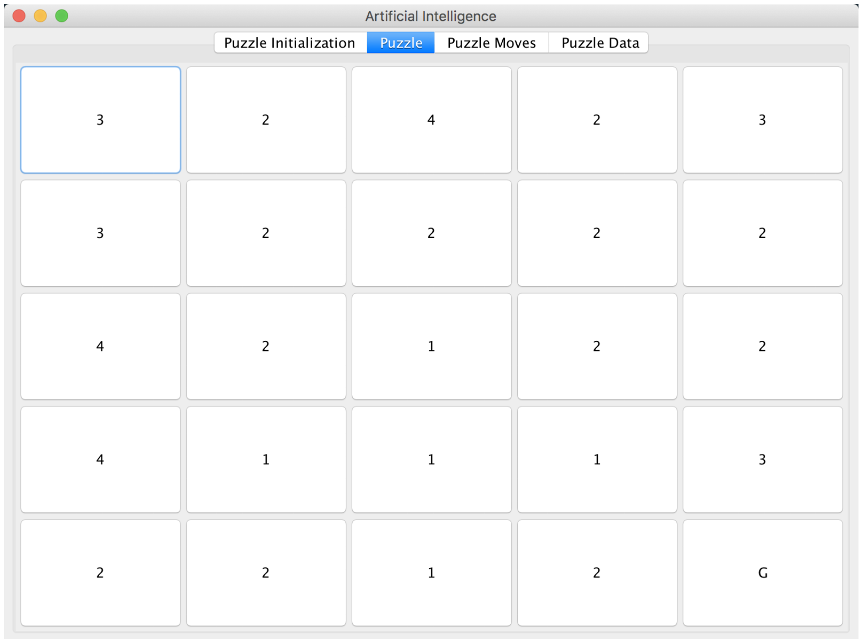

Figure 36: Basic Hill Climbing Puzzle for n = 5

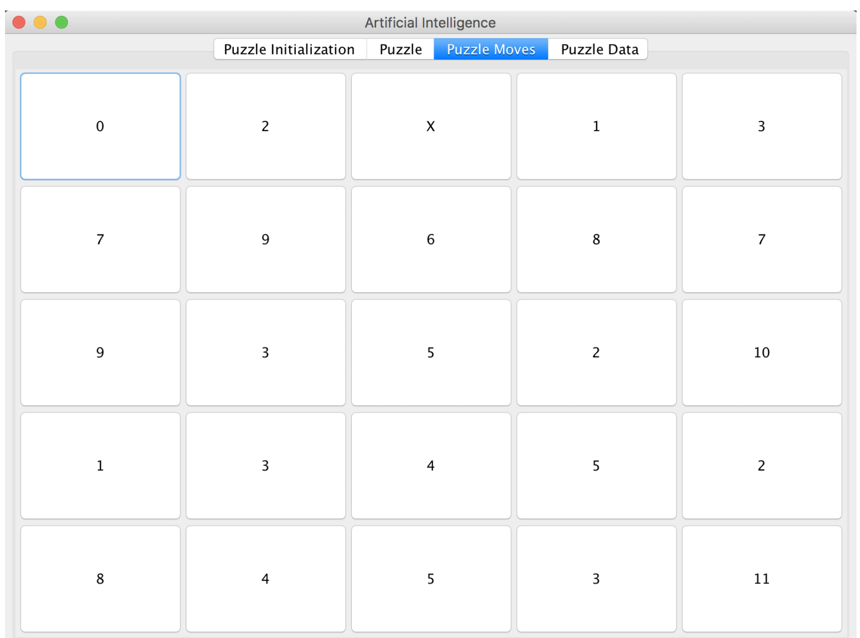

Figure 37: Basic Hill Climbing Puzzle Moves for n = 5

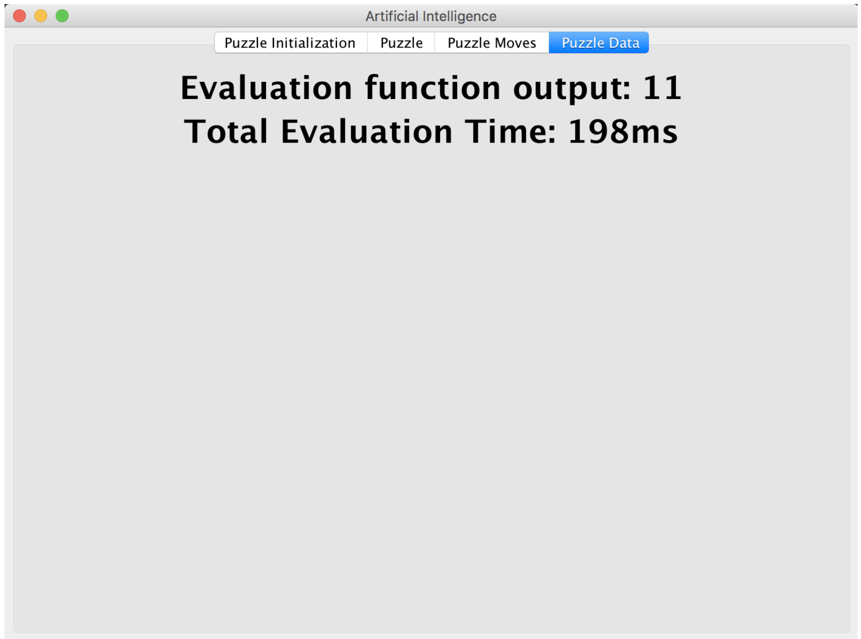

Figure 38: Basic Hill Climbing Puzzle Data for n = 5

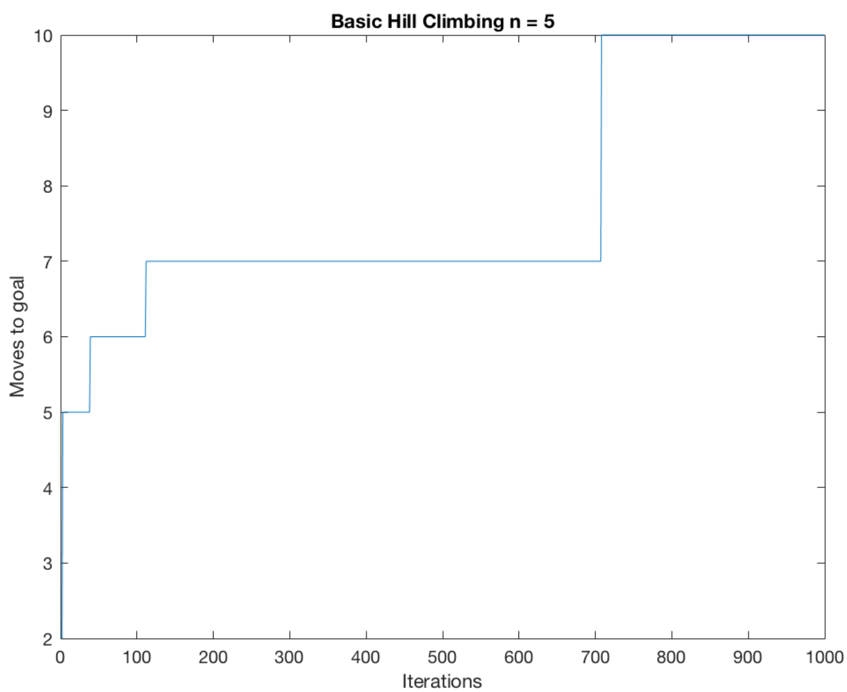

Figure 39: Plot of 1000 iterations of Hill Climbing for n = 5

Example Puzzle for n = 7

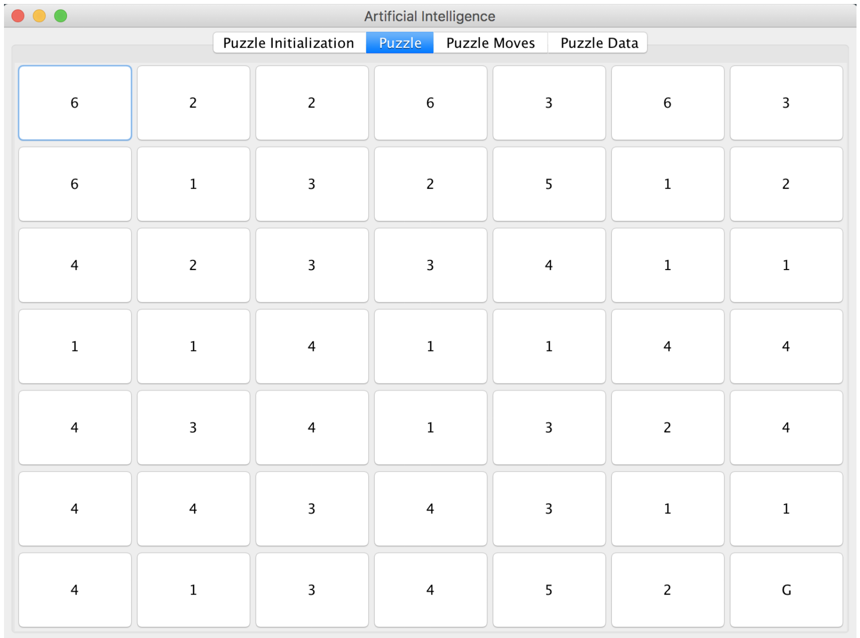

Figure 40: Basic Hill Climbing Puzzle for n = 7

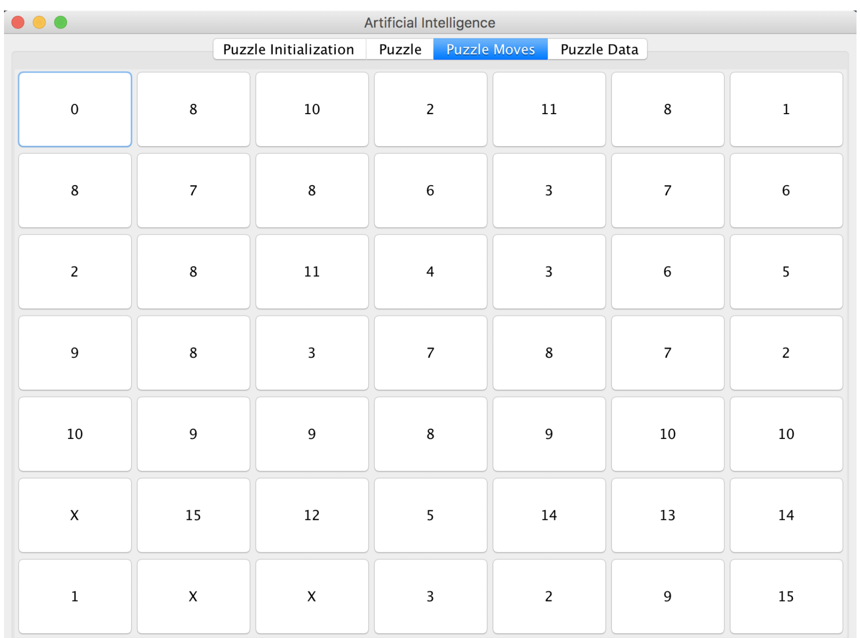

Figure 41: Basic Hill Climbing Puzzle Moves for n = 7

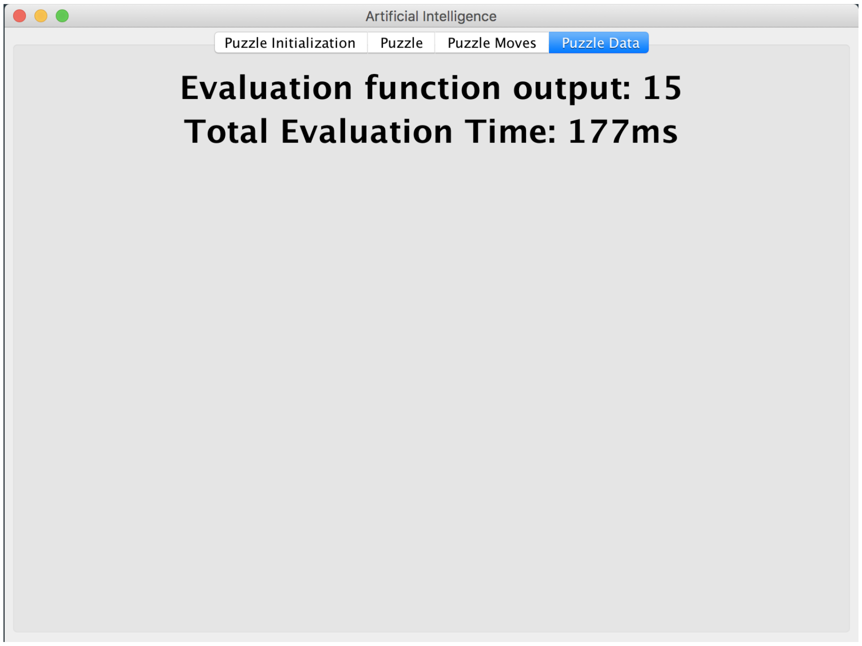

Figure 42: Basic Hill Climbing Puzzle Data for n = 7

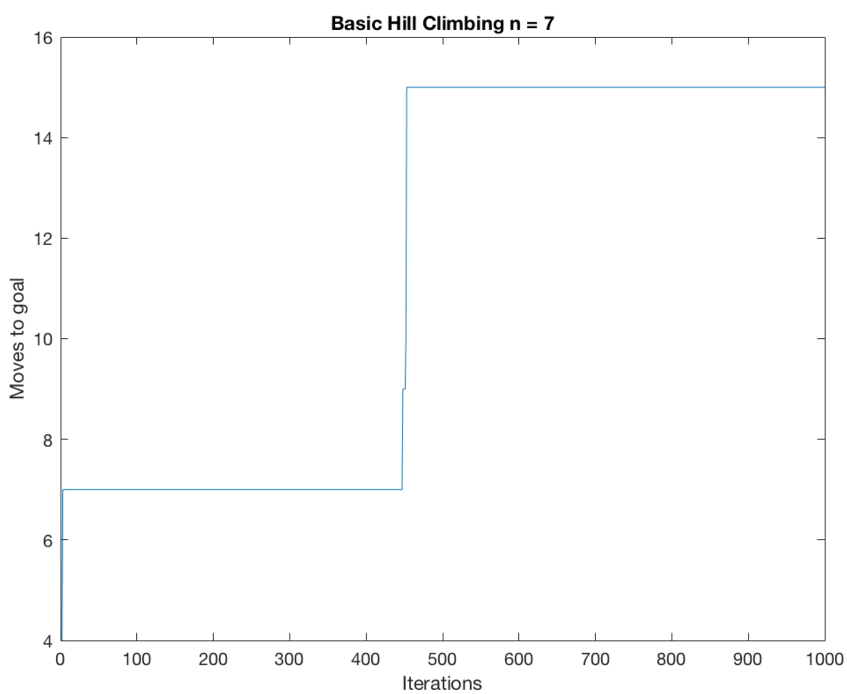

Figure 43: Plot of 1000 iterations of Hill Climbing for n = 7

Example Puzzle for n = 9

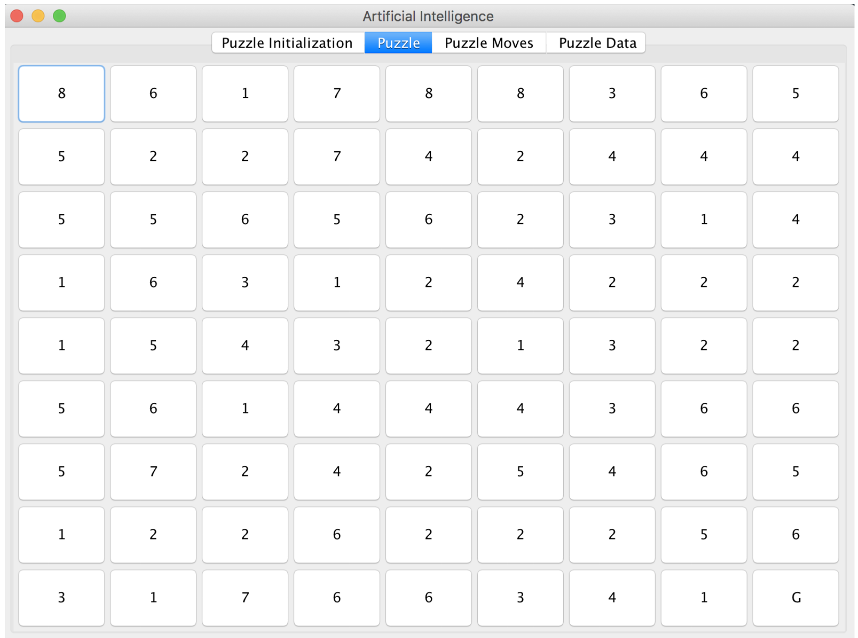

Figure 44: Basic Hill Climbing Puzzle for n = 9

Figure 45: Basic Hill Climbing Puzzle Moves for n = 9

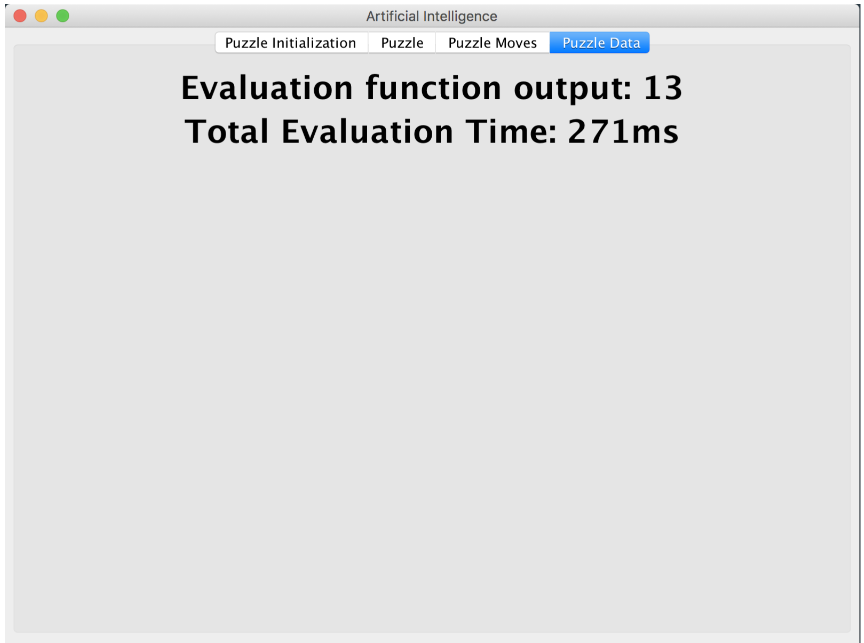

Figure 46: Basic Hill Climbing Puzzle Data for n = 9

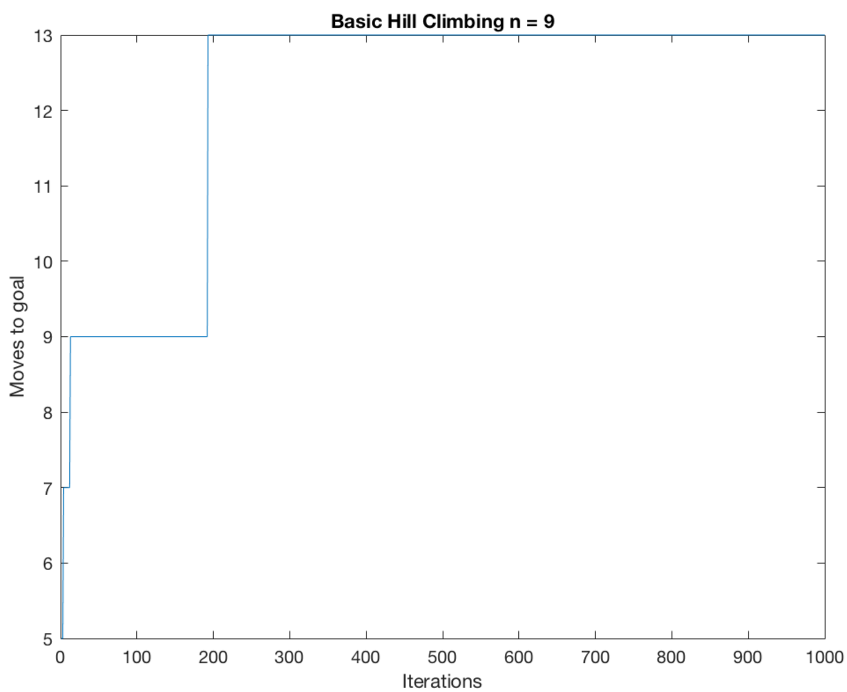

Figure 47: Plot of 1000 iterations of Hill Climbing for n = 9

Example Puzzle for n = 11

The first example is a "lucky" puzzle that started at a high number of moves. This is perhaps the best demonstration one of the problems of the stochastic hill climbing process. It shows how it can get stuck even after 1000 iterations in a local extrema. The run time is consequently much faster for this first puzzle as well since no other puzzles were found. The second puzzle also gets stuck at an extrema of 13 after 300 iterations that then lasts for 700 iterations.

Figure 48: Basic Hill Climbing Puzzle for n = 11

Figure 49: Basic Hill Climbing Puzzle Moves for n = 11

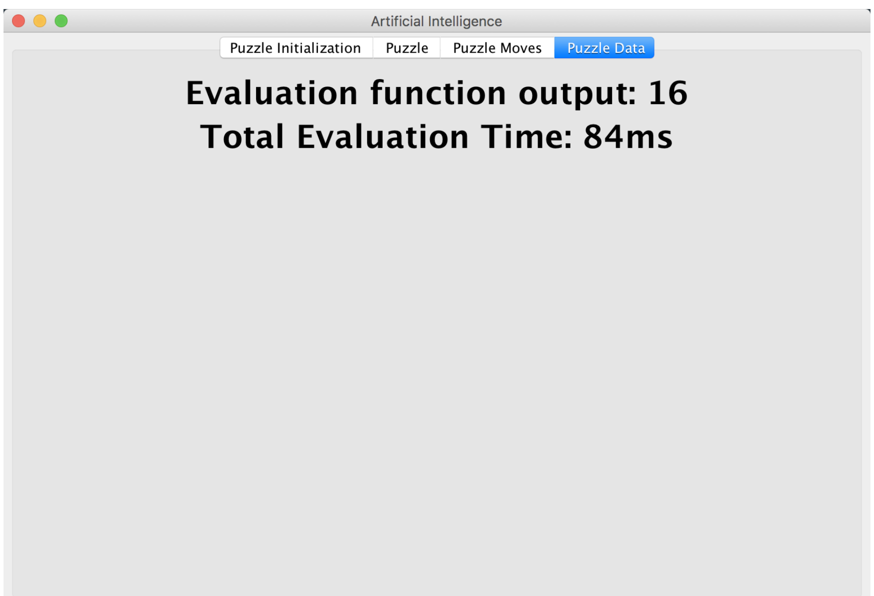

Figure 50: Basic Hill Climbing Puzzle Data for n = 11

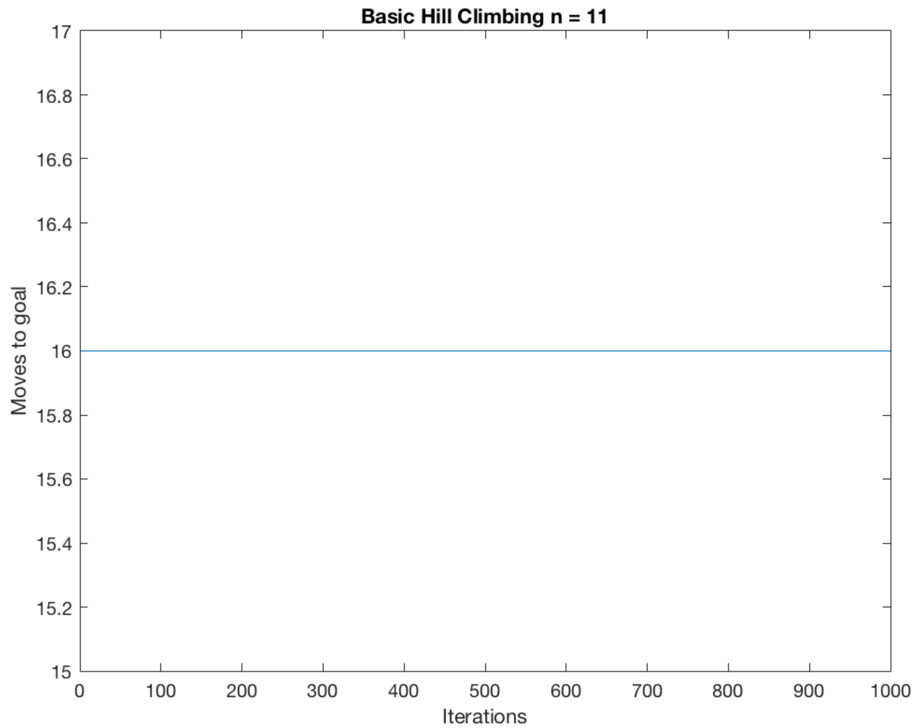

Figure 51: Plot of 1000 iterations of Hill Climbing for n = 11 of the first puzzle

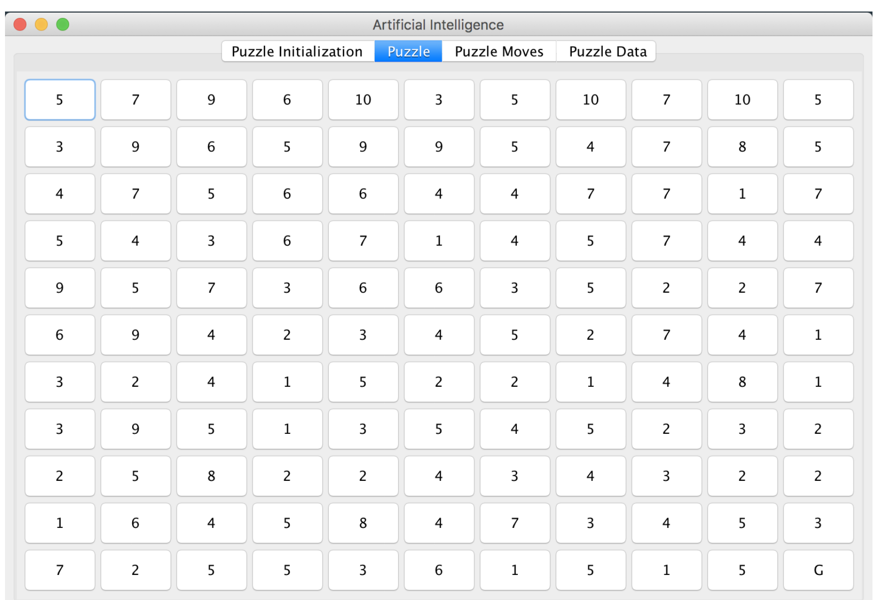

Figure 52: Basic Hill Climbing Puzzle 2 for n = 11

Figure 53: Basic Hill Climbing Puzzle 2 Moves for n = 11

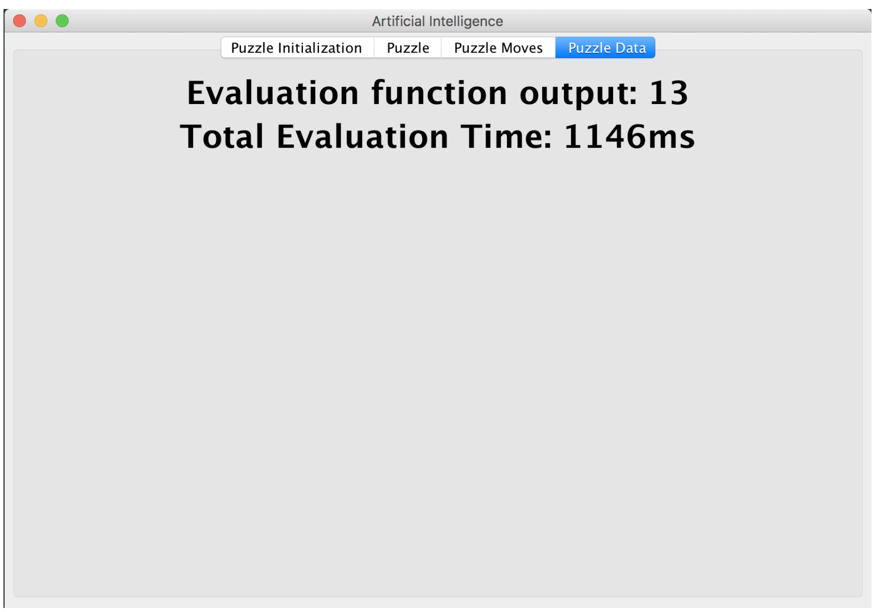

Figure 54: Basic Hill Climbing Puzzle 2 Data for n = 11

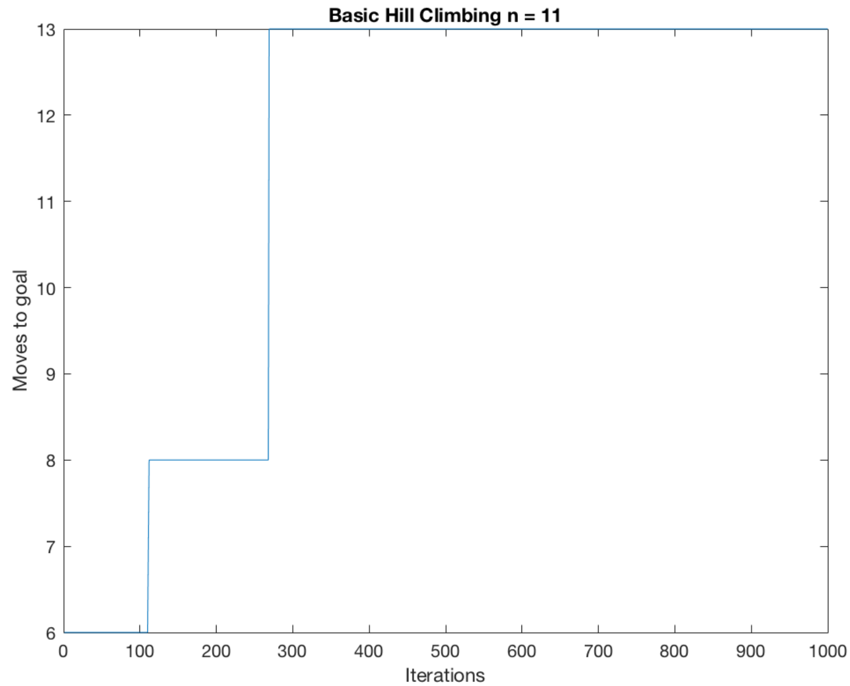

Figure 55: Plot of 1000 iterations of Hill Climbing for n = 11 of the second puzzle

Hill Climbing with Random Restarts

Your input in this case should be two numbers a) the number of times you will start a hill climbing process and b) the number of iterations per hill climbing process.

Example Puzzle for n = 5

IMAGES

Example Puzzle for n = 7

IMAGES

Example Puzzle for n = 9

IMAGES

Example Puzzle for n = 11

IIMAGES

Hill Climbing with Random Walks

Compare the output of the above two processes against the one that utilizes random walks for the same number of total iterations. i.e. again again visualize the final optimized puzzle configuration, its value and time it took to compute it.

Your input in this case will be two numbers a) the total number of iterations for hill climbing and b) the probability of the acceptance of a downhill move

Evaluate the effects of different values for probability p and select the one that works best for this problem and preferred number of total iterations

Example Puzzle for n = 5

IMAGES

Example Puzzle for n = 7

IMAGES

Example Puzzle for n = 9

IMAGES

Example Puzzle for n = 11

IIMAGES

Simulated Annealing

Proposal and Implementation of a population based approach