AI Assignment 1, CS 440

Kelvin Liu

September 27, 2017

1 Task 1: Puzzle Representation

1.1 Examples of GUI

Initial Board The following two puzzles are a randomly generated puzzle and its corresponding distance board, where the numbers correspond to the minimum number of jumps need to get from the top left cell to that cell. text here

Initial Board

Puzzle Board

	1	2	3	4	5	6	7
1	5	4	4	2	3	3	6
2	3	3	2	5	4	1	4
3	3	3	2	2	1	5	5
4	6	4	1	3	3	1	6
5	6	5	2	1	4	2	6
6	3	4	3	2	4	5	5
7	6	2	6	2	2	1	0

Show Corresponding Distance/Puzzle Board

Score: 4

Initial Board

Distance Board

	1	2	3	4	5	6	7
1	0	5	2	4	4	1	3
2	6	4	-1	7	5	-1	-1
3	2	4	4	3	5	3	-1
4	4	4	-1	3	3	2	3
5	4	5	3	4	4	3	5
6	1	3	-1	2	6	3	-1
7	-1	5	4	4	4	4	4

Show Corresponding Distance/Puzzle Board

Score: 4

Final Board These next picture is of the previous puzzle state after a basic hill climbing routine. The picture after that is the corresponding distance board.

Final Board

Puzzle Board

	1	2	3	4	5	6	7
1	5	4	2	5	1	5	6
2	2	4	2	4	3	2	3
3	3	2	2	1	4	1	6
4	1	5	1	2	3	2	4
5	2	3	1	4	4	4	1
6	3	3	5	2	3	5	1
7	5	3	2	3	3	1	0

Show Corresponding Distance/Puzzle Board

Score: 10

Final Board

Distance Board

	1	2	3	4	5	6	7
1	0	5	5	7	6	1	-1
2	9	6	10	4	7	5	-1
3	2	4	4	3	4	-1	-1
4	10	4	6	3	6	4	5
5	7	5	5	6	6	-1	-1
6	1	3	6	2	4	2	-1
7	8	6	-1	-1	5	9	10

Show Corresponding Distance/Puzzle Board

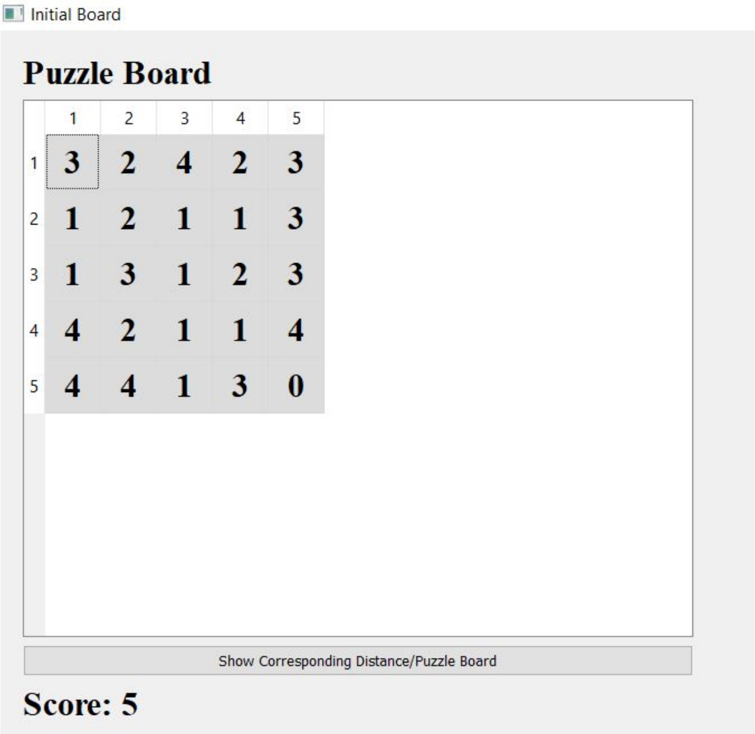
Score: 10

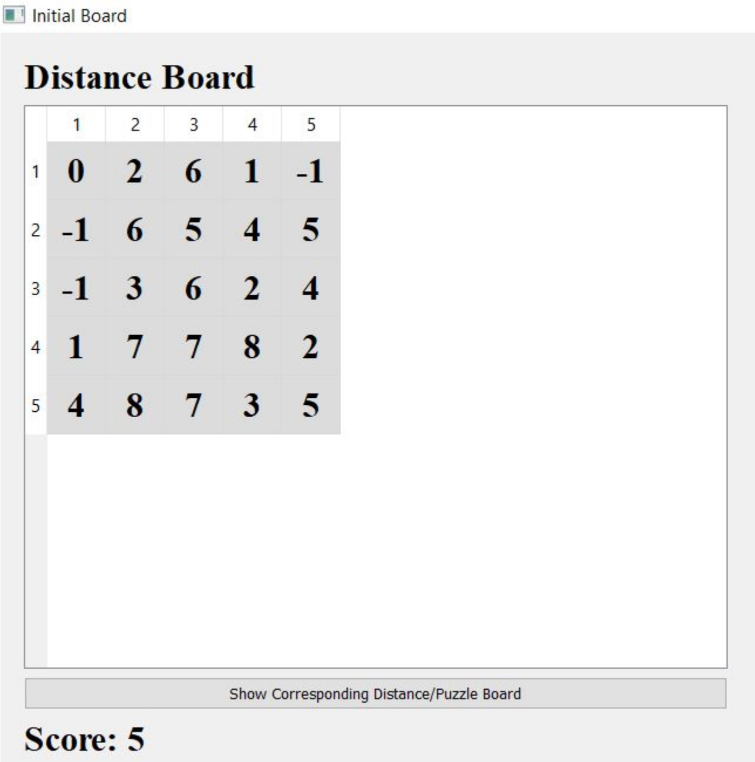
2 Task 2: Puzzle Evaluation

The following are examples of solvable and unsolvable boards for $n = 5, 7, 9, 11$. After each puzzle is the distance board.

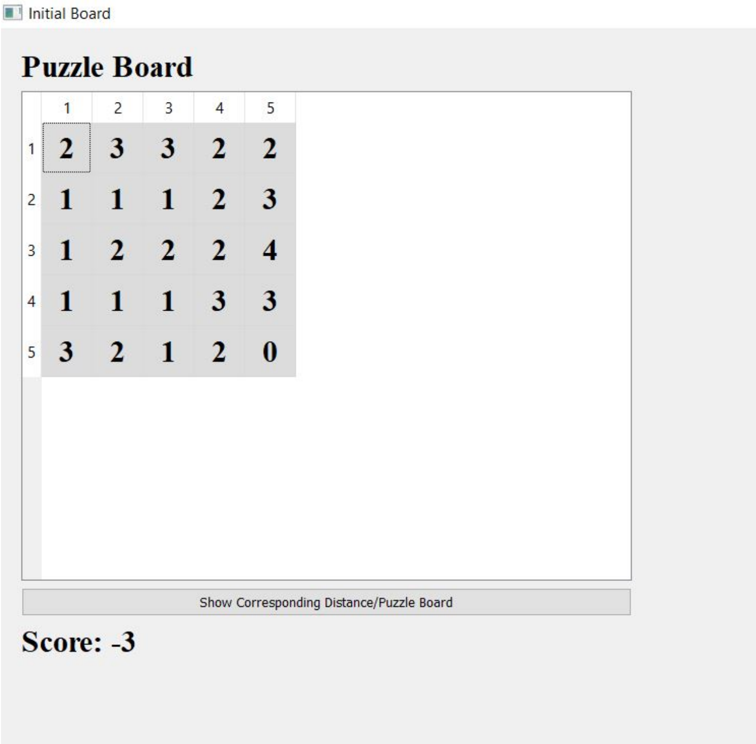
2.1 Size 5 Boards

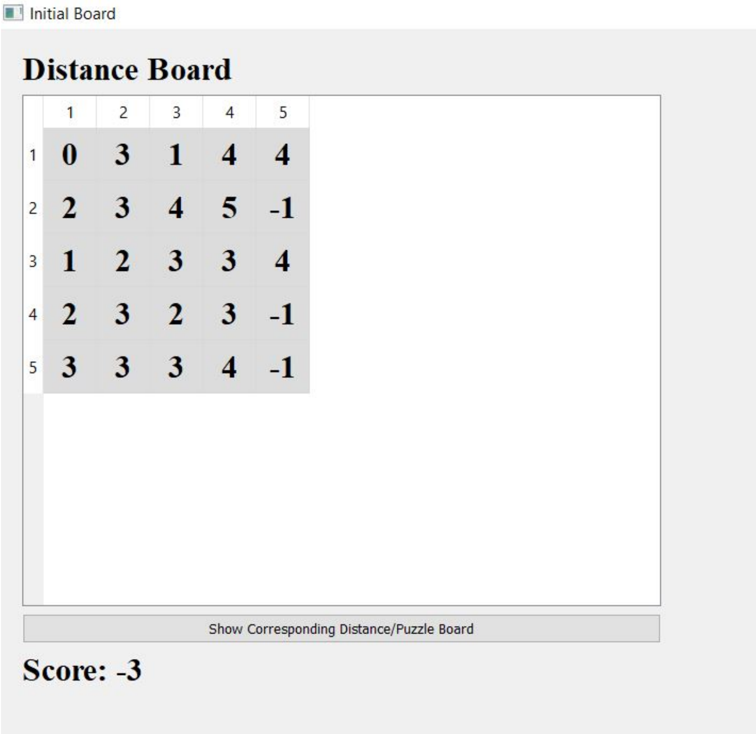
Solvable





Unsolvable


Initial Board
Puzzle Board
1 2 3 4 5
1 2 3 3 2 2
2 1 1 1 2 3
3 1 2 2 2 4
4 1 1 1 3 3
5 3 2 1 2 0
Show Corresponding Distance/Puzzle Board
Score: -3


Initial Board
Distance Board
1 2 3 4 5
1 0 3 1 4 4
2 2 3 4 5 -1
3 1 2 3 3 4
4 2 3 2 3 -1
5 3 3 3 4 -1
Show Corresponding Distance/Puzzle Board
Score: -3

2.2 Size 7 Boards

Solvable

Initial Board

Puzzle Board

	1	2	3	4	5	6	7
1	6	6	3	5	4	1	6
2	5	2	2	5	5	1	6
3	5	5	1	3	1	3	3
4	4	4	3	3	1	4	3
5	1	1	2	1	4	2	2
6	6	2	1	5	1	2	4
7	4	4	4	3	2	5	0

Show Corresponding Distance/Puzzle Board

Score: 2

Initial Board

Distance Board

	1	2	3	4	5	6	7
1	0	-1	6	6	4	7	1
2	6	-1	5	-1	6	7	7
3	2	5	4	5	7	3	6
4	5	6	5	7	6	5	8
5	4	5	6	-1	3	-1	-1
6	5	6	7	5	-1	4	6
7	1	-1	3	8	2	-1	2

Show Corresponding Distance/Puzzle Board

Score: 2

Unsolvable

Initial Board

Puzzle Board

	1	2	3	4	5	6	7
1	1	5	2	5	2	6	3
2	3	2	2	5	3	1	6
3	3	5	1	2	3	1	6
4	6	1	3	2	4	5	1
5	6	2	1	1	2	4	4
6	5	3	5	5	5	3	3
7	1	3	3	1	6	4	0

Show Corresponding Distance/Puzzle Board

Score: -5

Initial Board

Distance Board

	1	2	3	4	5	6	7
1	0	1	5	3	4	8	2
2	1	9	7	2	8	7	8
3	5	3	6	6	5	6	4
4	5	7	5	6	-1	4	3
5	2	5	4	5	6	-1	3
6	6	2	5	4	3	7	-1
7	-1	6	4	3	4	5	-1

Show Corresponding Distance/Puzzle Board

Score: -5

2.3 Size 9 Boards

Solvable

Initial Board

Puzzle Board

	1	2	3	4	5	6	7	8	9
1	8	5	5	3	3	5	3	2	3
2	6	4	3	4	4	1	6	2	4
3	5	2	3	2	5	2	4	2	2
4	6	7	4	5	4	2	3	5	8
5	5	4	6	3	1	1	4	4	2
6	7	6	5	5	1	2	4	4	6
7	7	1	6	1	2	6	1	5	8
8	5	7	3	2	4	4	1	4	4
9	6	5	7	5	7	2	8	3	0

Show Corresponding Distance/Puzzle Board

Score: 7

Initial Board

Distance Board

	1	2	3	4	5	6	7	8	9
1	0	6	7	4	-1	2	3	5	1
2	7	-1	-1	6	6	5	6	5	7
3	2	5	10	4	-1	3	9	4	8
4	3	8	6	5	6	4	4	5	2
5	6	6	6	5	5	4	5	5	7
6	-1	7	6	4	6	3	6	4	5
7	9	-1	7	-1	7	6	5	6	8
8	3	5	7	6	7	4	6	7	6
9	1	7	-1	6	7	-1	2	6	7

Show Corresponding Distance/Puzzle Board

Score: 7

Unsolvable

Initial Board

Puzzle Board

	1	2	3	4	5	6	7	8	9
1	1	8	5	3	6	6	6	4	2
2	5	1	2	5	4	6	3	3	4
3	8	6	5	5	4	3	1	2	3
4	8	3	3	1	4	5	6	3	8
5	8	4	6	2	1	1	3	5	7
6	5	7	4	2	5	4	4	2	2
7	8	5	1	6	2	5	5	4	4
8	6	7	6	2	2	3	7	5	5
9	2	6	5	8	3	7	5	2	0

Show Corresponding Distance/Puzzle Board

Score: -20

Initial Board

Distance Board

	1	2	3	4	5	6	7	8	9
1	0	1	7	6	7	-1	7	5	-1
2	1	7	5	7	6	2	6	-1	7
3	-1	3	6	7	-1	5	-1	4	4
4	6	-1	6	7	6	5	-1	-1	6
5	-1	7	6	6	5	4	5	5	7
6	-1	6	6	6	6	5	6	-1	5
7	2	6	5	5	4	-1	5	4	3
8	-1	6	4	5	7	3	6	-1	4
9	-1	2	-1	-1	5	4	-1	3	-1

Show Corresponding Distance/Puzzle Board

Score: -20

2.4 Size 11 Boards

Solvable

Initial Board

Puzzle Board

	1	2	3	4	5	6	7	8	9	10	11
1	5	5	1	1	5	7	10	1	4	10	3
2	4	4	1	8	5	3	4	6	2	2	1
3	7	4	1	5	7	4	5	1	4	9	10
4	1	6	6	6	3	7	1	6	3	6	9
5	5	5	1	7	1	2	6	3	3	9	9
6	8	6	7	6	6	3	3	7	2	4	8
7	4	4	7	7	1	1	6	6	1	2	3
8	2	5	5	3	3	3	2	7	2	5	7
9	6	2	5	7	2	5	4	3	8	3	10
10	2	2	7	1	1	7	1	3	5	5	6
11	6	2	4	1	6	1	1	4	4	10	0

Show Corresponding Distance/Puzzle Board

Score: 5

Initial Board

Distance Board

	1	2	3	4	5	6	7	8	9	10	11
1	0	7	6	7	5	1	6	5	4	6	5
2	-1	6	5	6	-1	7	7	5	6	5	7
3	6	5	4	5	7	4	4	7	6	5	8
4	-1	7	5	7	6	4	-1	7	3	6	6
5	6	6	-1	4	5	3	5	4	5	6	5
6	1	7	4	4	6	5	3	7	2	4	3
7	6	6	8	6	5	4	5	5	4	5	7
8	5	6	3	5	5	2	4	4	3	-1	4
9	6	8	5	6	7	6	4	6	-1	6	5
10	6	7	6	5	6	4	5	6	4	5	7
11	7	7	7	6	4	3	4	5	-1	7	5

Show Corresponding Distance/Puzzle Board

Score: 5

Unsolvable

Initial Board

Puzzle Board

	1	2	3	4	5	6	7	8	9	10	11
1	5	2	10	4	8	8	1	10	3	6	7
2	1	6	6	2	4	9	2	2	5	9	10
3	2	7	5	3	7	7	1	6	3	9	4
4	5	1	8	6	4	2	3	3	2	2	8
5	6	8	1	7	5	5	4	6	8	5	4
6	4	5	2	4	2	5	5	1	4	4	1
7	2	3	7	1	2	2	3	6	1	6	10
8	1	4	6	7	4	7	5	5	2	3	4
9	3	2	7	5	8	4	8	8	1	2	4
10	9	5	3	5	6	4	7	6	1	3	10
11	6	10	7	8	8	4	6	7	3	2	0

Show Corresponding Distance/Puzzle Board

Score: -16

Initial Board

Distance Board

	1	2	3	4	5	6	7	8	9	10	11
1	0	5	-1	6	7	1	4	5	-1	5	5
2	2	3	5	6	6	5	5	4	5	5	-1
3	3	6	4	-1	-1	6	5	5	7	-1	-1
4	4	5	4	5	3	5	5	5	4	6	5
5	4	6	6	7	6	3	5	8	-1	7	4
6	1	4	3	6	2	5	3	7	5	7	6
7	5	4	6	5	5	7	6	6	6	4	7
8	4	4	4	6	3	5	5	-1	4	-1	5
9	5	3	-1	4	6	2	6	4	5	3	5
10	2	5	-1	6	7	4	4	6	5	3	8
11	5	4	-1	-1	-1	6	4	5	6	4	-1

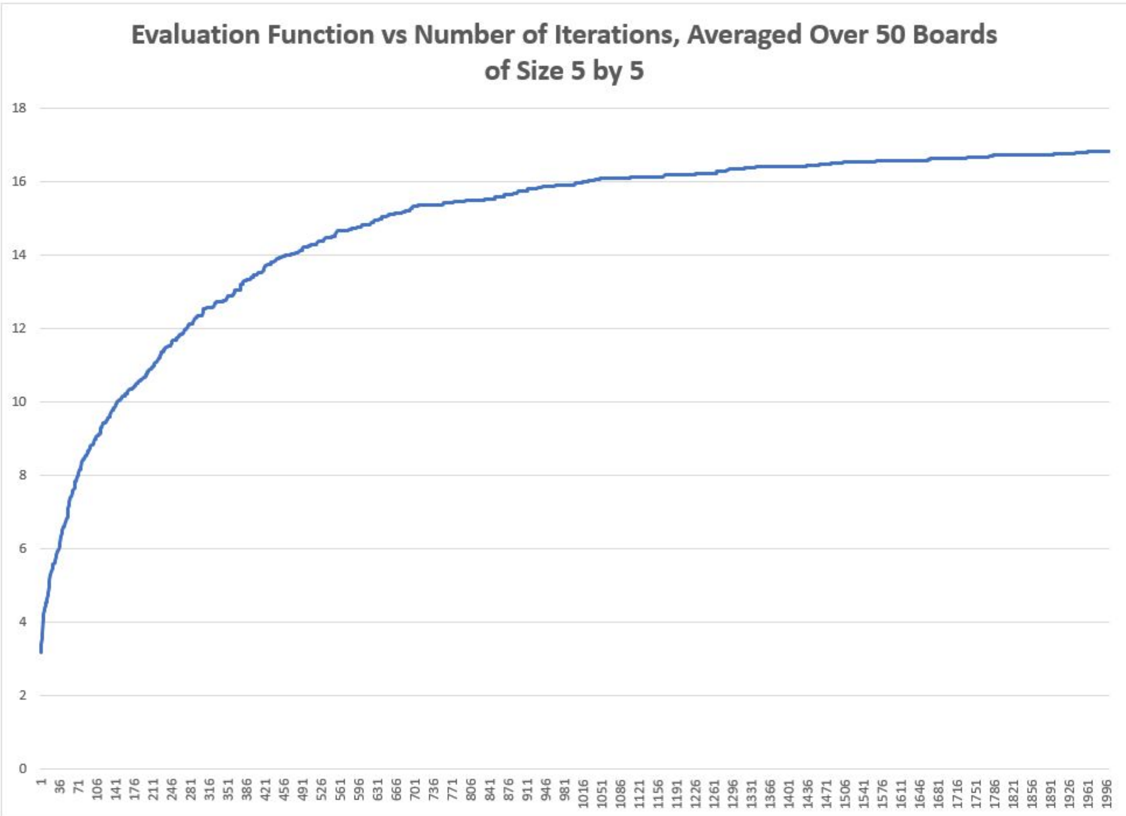
Show Corresponding Distance/Puzzle Board

Score: -16

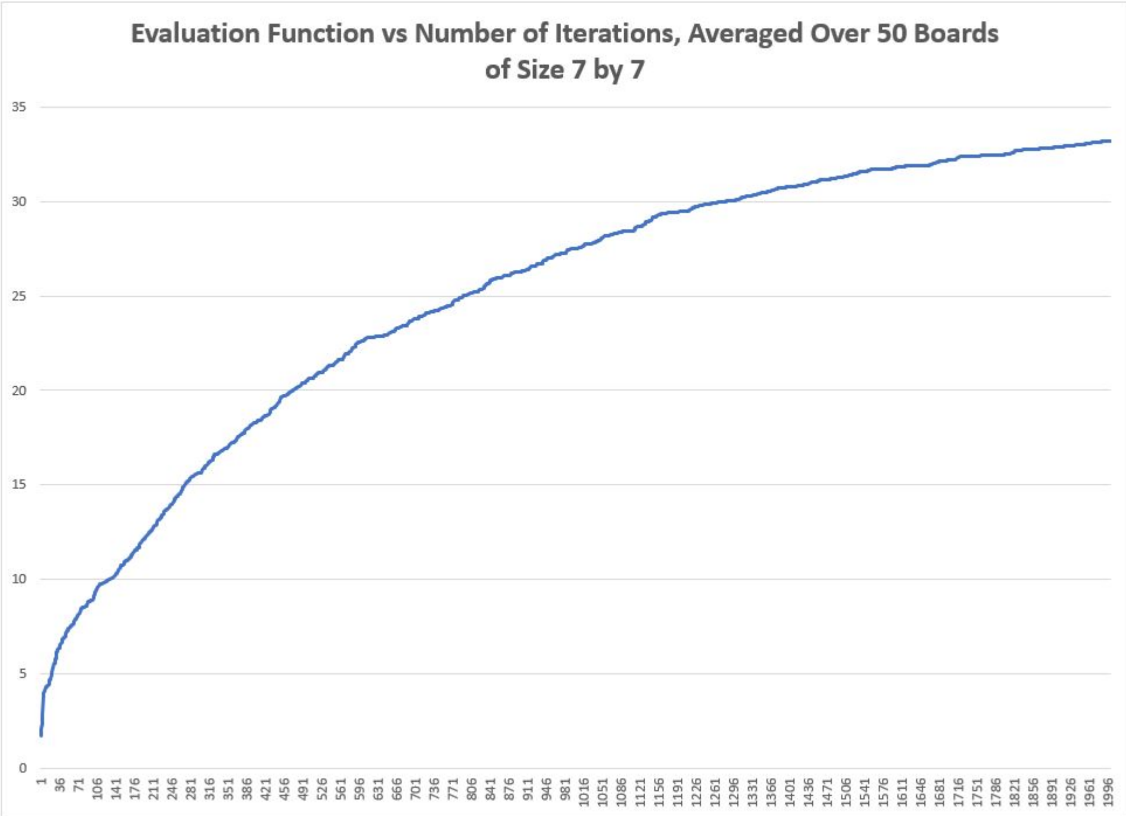
3 Task 3: Basic Hill Climbing Approach

To analyze the effect of hill climbing on the evaluation function, the following graphs are supplied. For a given board of size n, each plot shows the increase of the evaluation function with respect to the number of iterations performed, up to 2000 iterations. Note that in each case the evaluation function seems to be increasing towards an asymptote.

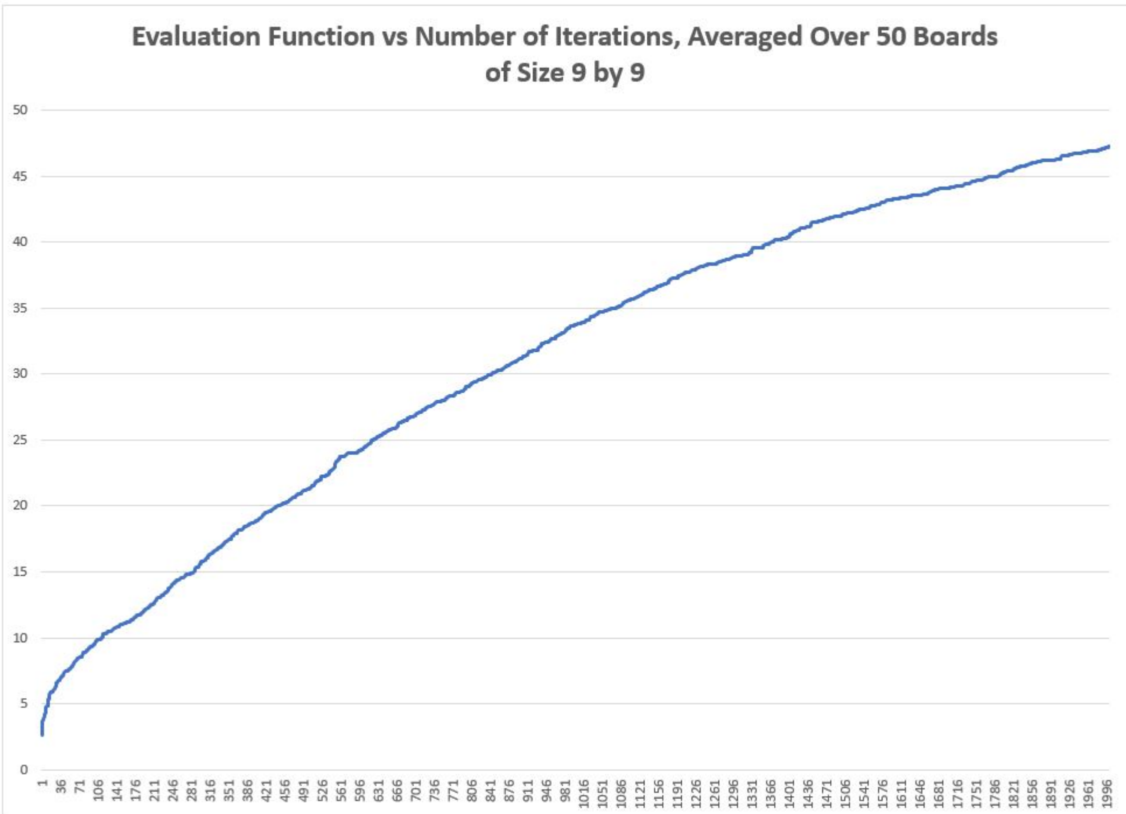
3.1 Size 5 Boards



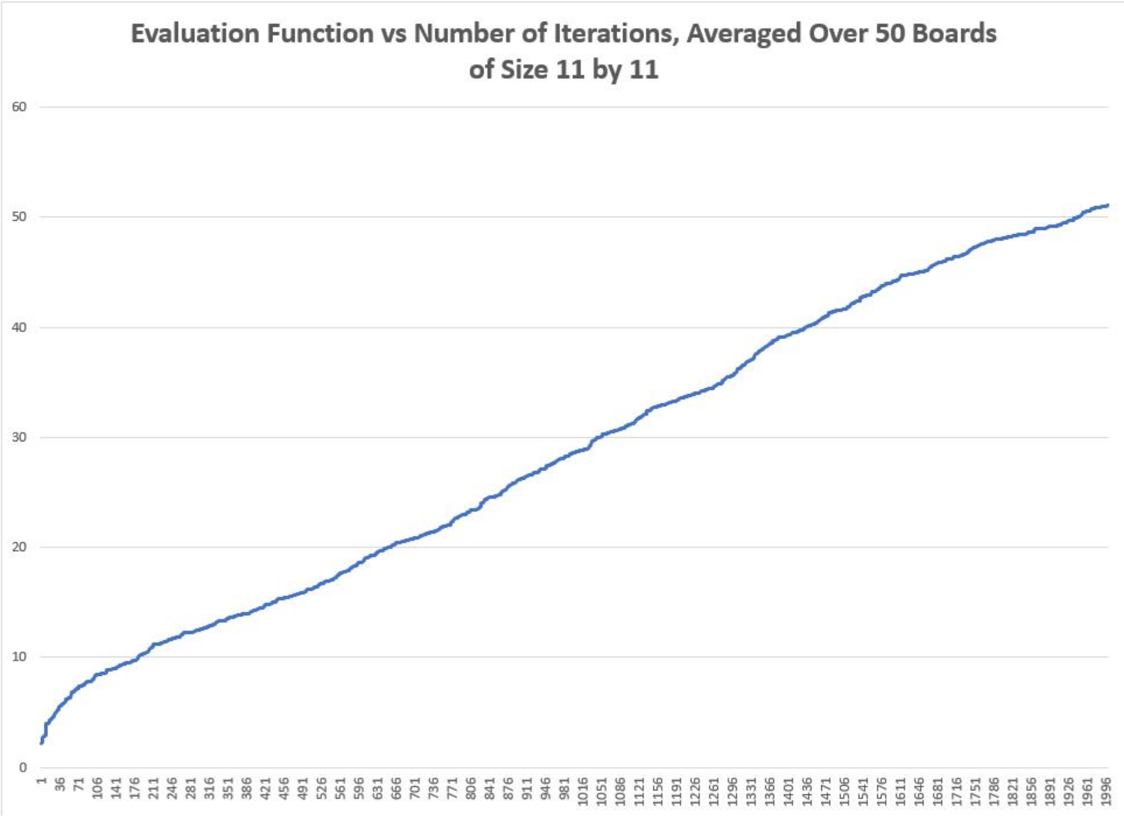
3.2 Size 7 Boards



3.3 Size 9 Boards



3.4 Size 11 Boards

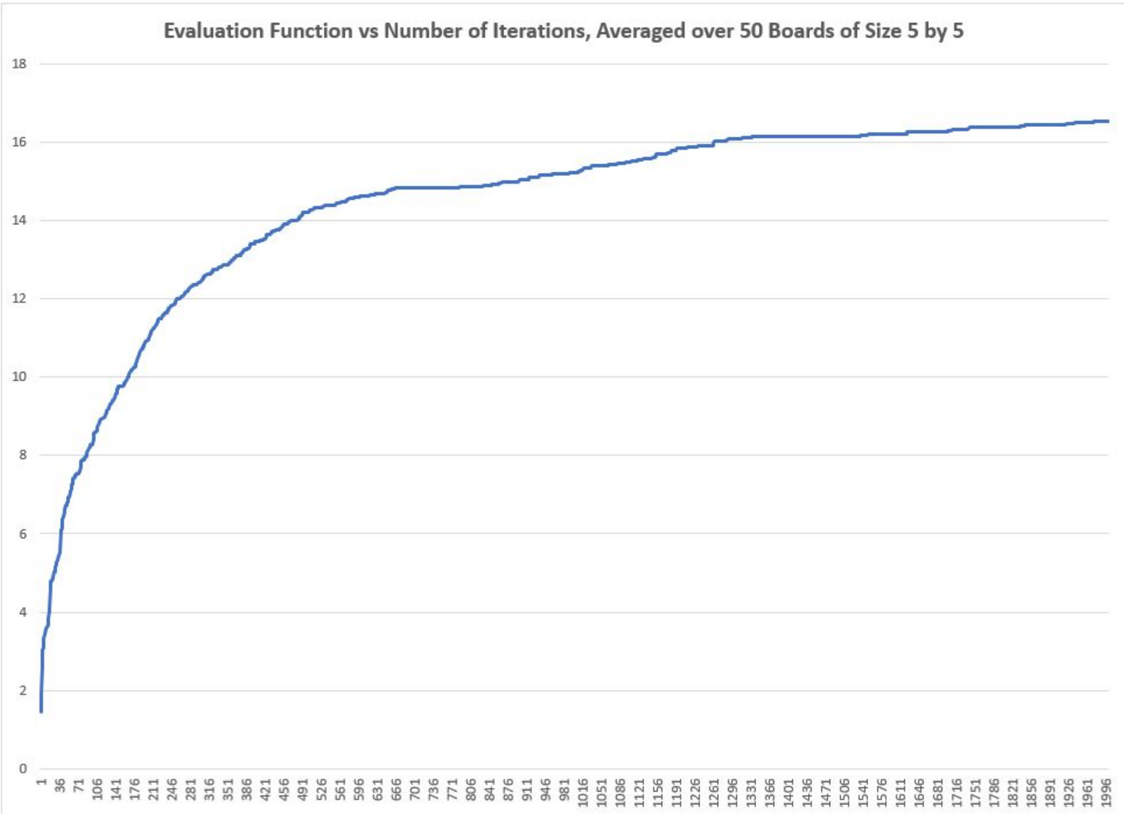


4 Task 4: Hill Climbing with Random Restarts

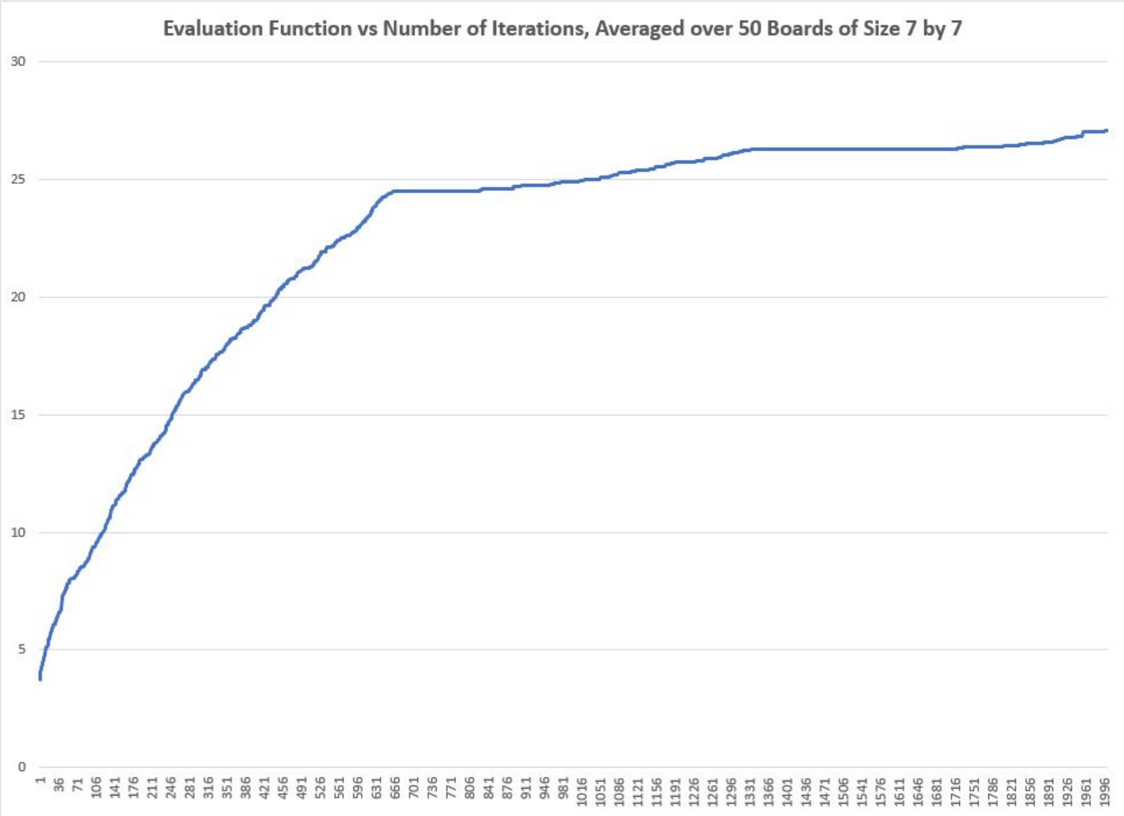
4.1 Comparison to Basic Hill Climbing

Here we introduce random restarts into the algorithm, keeping the number of iterations the same at 2000. Note how because we are taking best values when doing restarts, this graph is much less "smooth" than the basic hill climbing graph. They both reach around the same values for the smaller boards, though for the larger boards restarting lowers the evaluation function because there are not enough iterations done to reach an equilibrium if the process restarts. Due to concerns with processing speed, I chose to restart 3 times throughout.

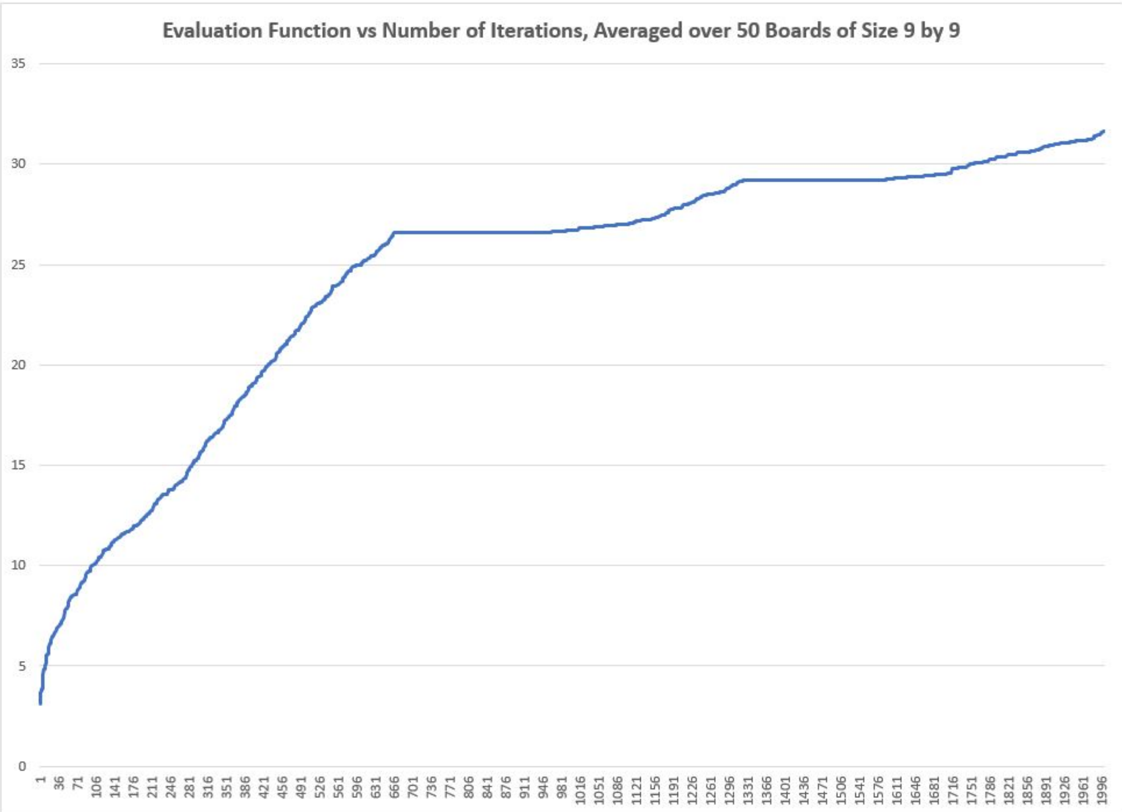
4.2 Size 5 Boards



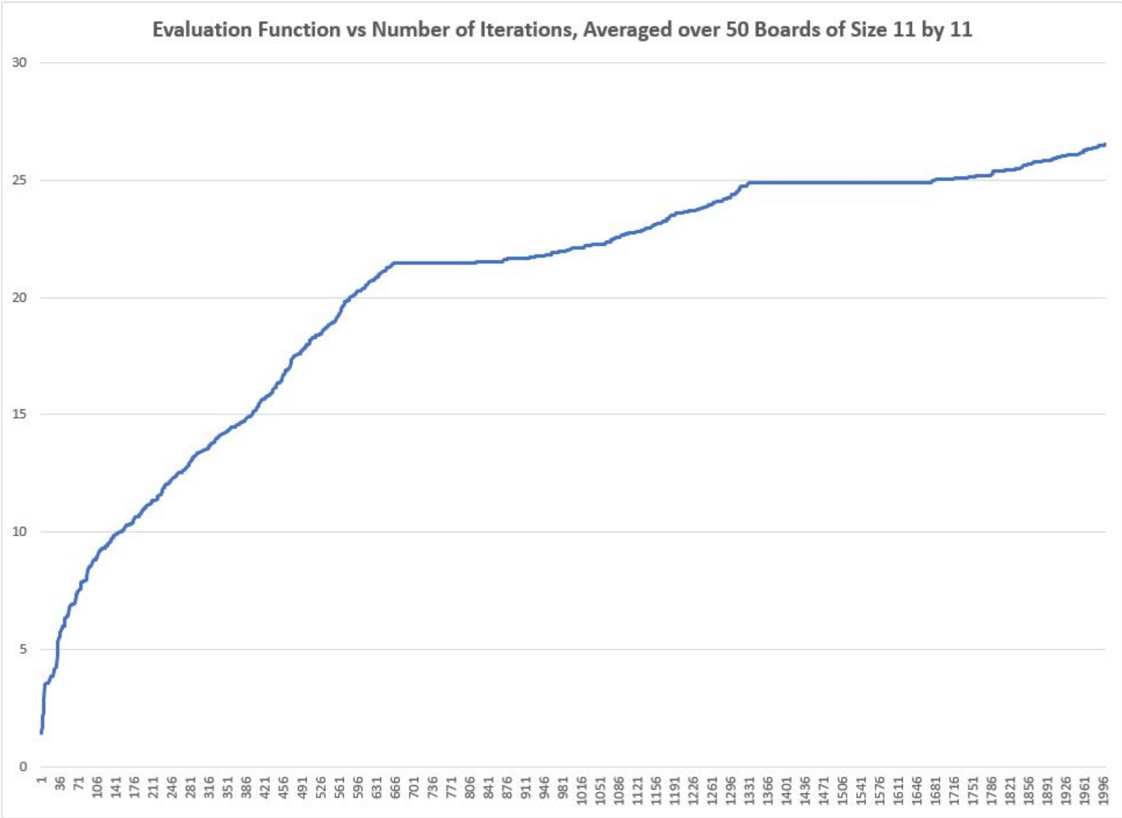
4.3 Size 7 Boards



4.4 Size 9 Boards



4.5 Size 11 Boards



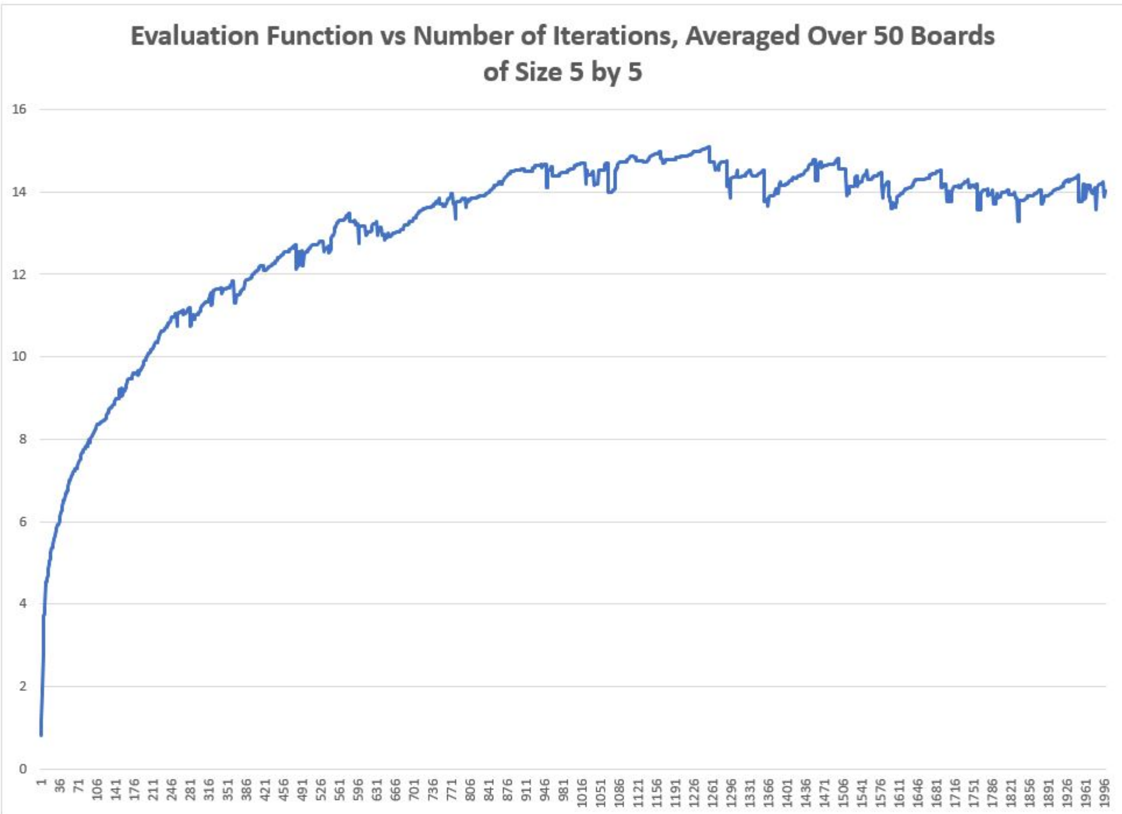
5 Task 5: Hill Climbing with Random Walk

Comparison to Basic Hill Climbing and Random Restart

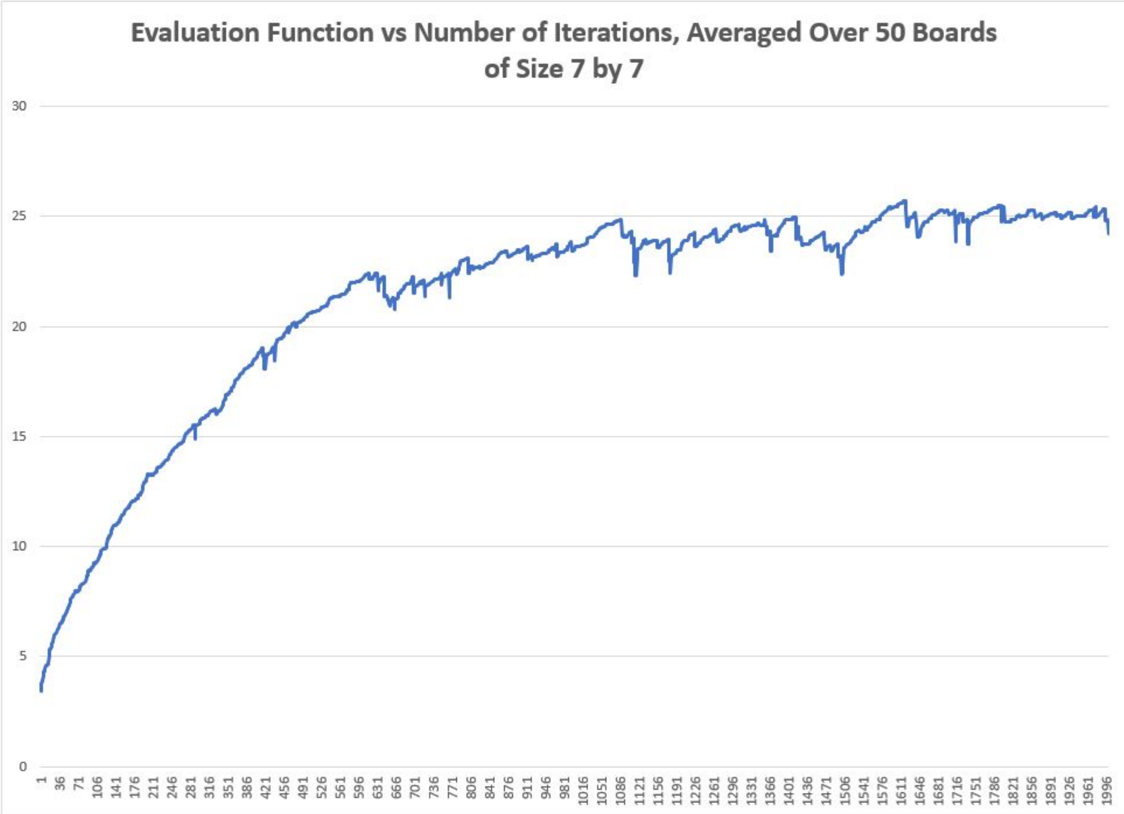
By introducing a chance for "bad" moves to be accepted, the evaluation function still trends upwards with the number of iterations, but in a neighborhood of a point, there may be many more jumps corresponding to mutations having a large, sudden negative impact on a puzzle. Also note that the function jumps around more, or is more discontinuous, even after the evaluation function reaches its equilibrium. I hypothesize that this is due to the low likelihood that a random change will produce a better board once most optimizing changes have been made. Therefore any random change would have more potential to be harmful, since there is a non-negative chance, p, of being accepted.

Once all the other parameters were set, I did quick searches for the best value of p, which was around 0.002. For p values higher than 0.1, the evaluation function becomes very erratic and does not reliably reach equilibrium.

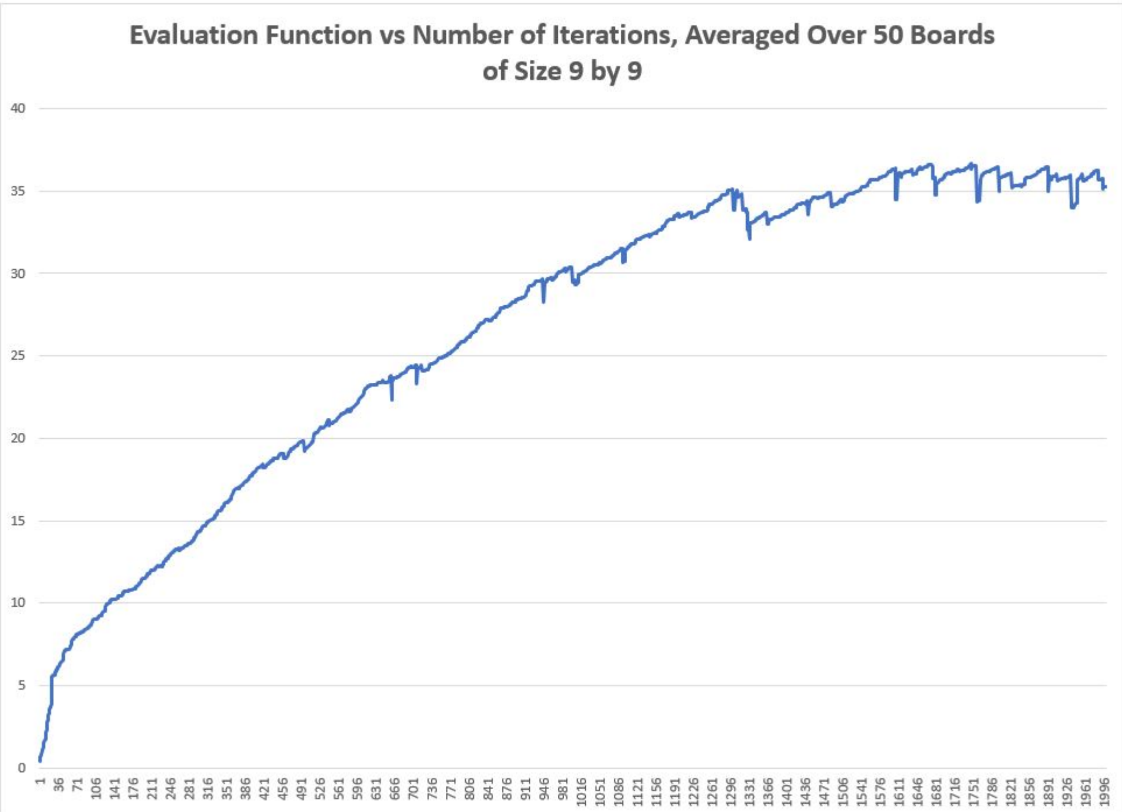
5.1 Size 5 Boards



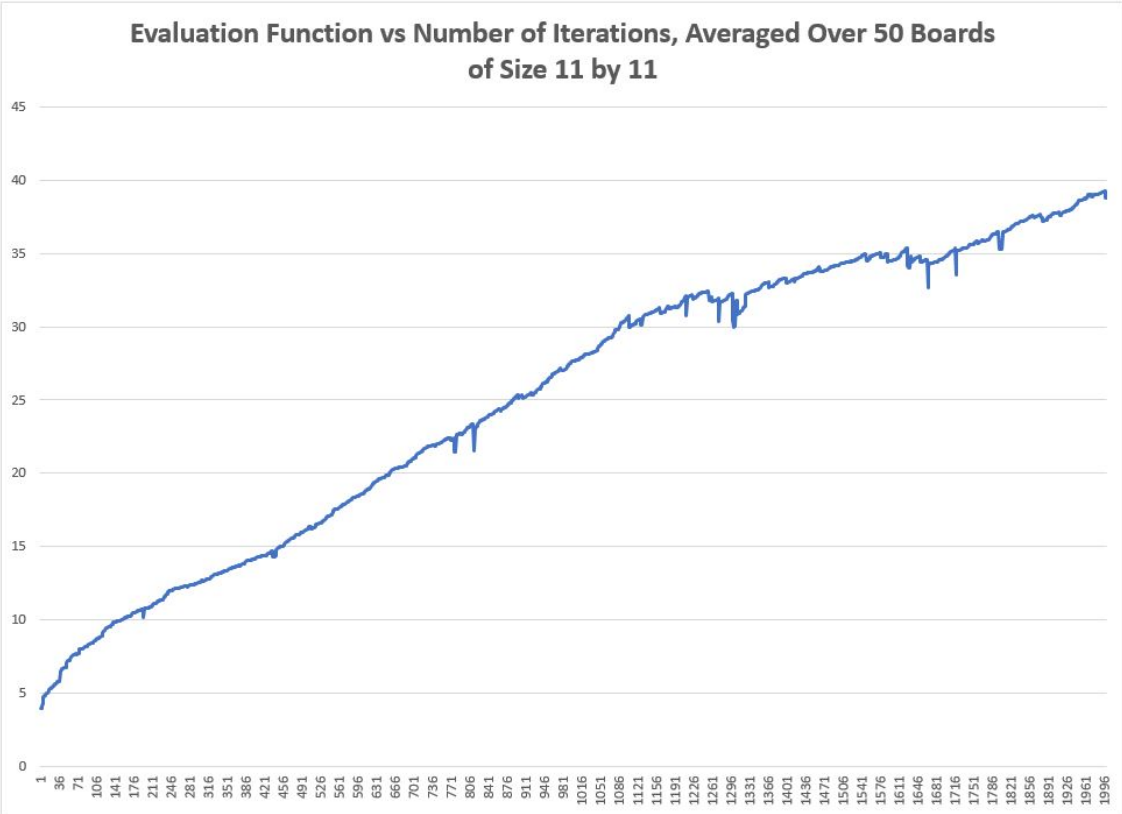
5.2 Size 7 Boards



5.3 Size 9 Boards



5.4 Size 11 Boards



6 Task 6: Simulated Annealing

Comparison to Hill Climb, Random Restart, Random Walk

One issue with a simple random walk strategy is that once the evaluation of the puzzles reaches an "upper limit" as shown in the graphs, the likelihood of a downward step is more likely and does not help increase the evaluation function much. Another issue, as mentioned in the problem, is that small bad steps are equally likely as large bad steps. We wish to make large bad steps more unlikely than small ones.

With simulated annealing, a temperature and decay parameter ensure that the evaluation function plot will stably tend towards the asymptote. A further benefit is that at the beginning of the puzzle iterations, a large temperature allows many "bad" moves to be made, which spread out the puzzles over the state space. This can be seen visually on the graph–at low iteration numbers the evaluation function average jumps around erratically.

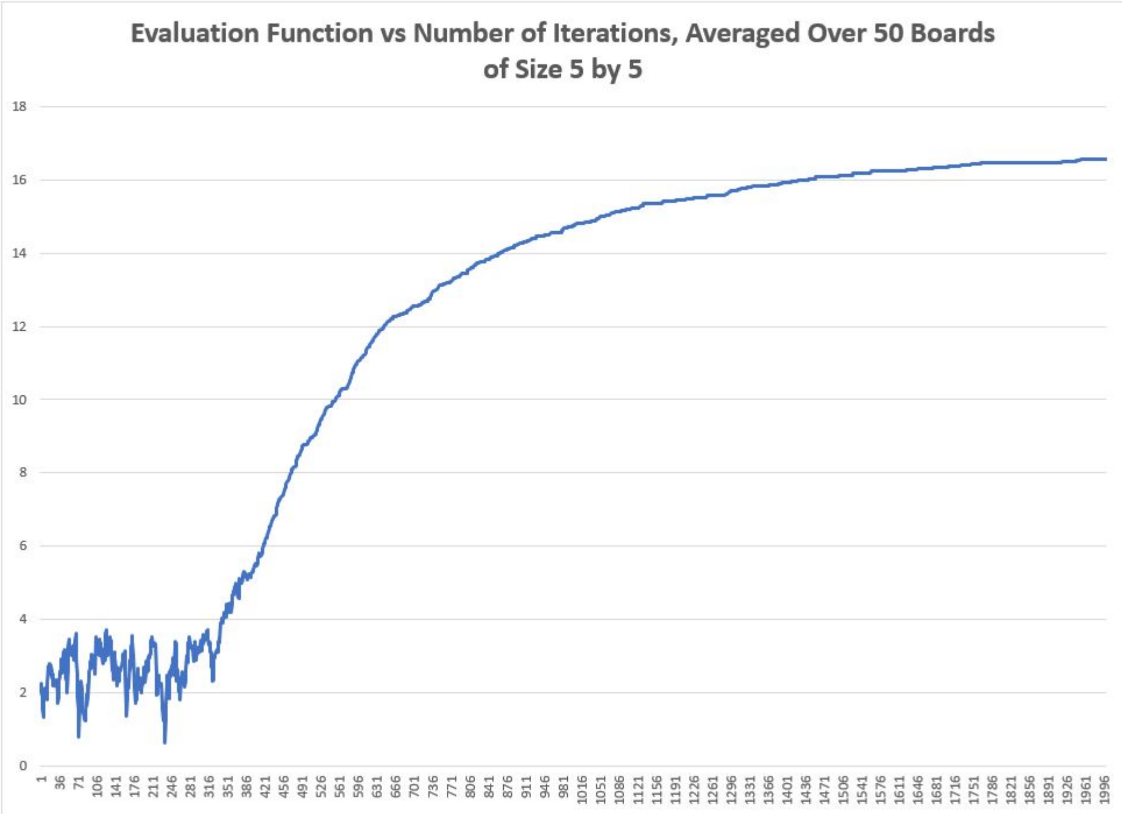
For these graphs, I used $T = 25000$ and $d = 0.975$. T was chosen to be very large as to allow for lots of time for random walk in the beginning. d was chosen to be very close to 1 and such that

$$T \times d^{\text{iterations}/10} \not\approx 0 \tag{1}$$

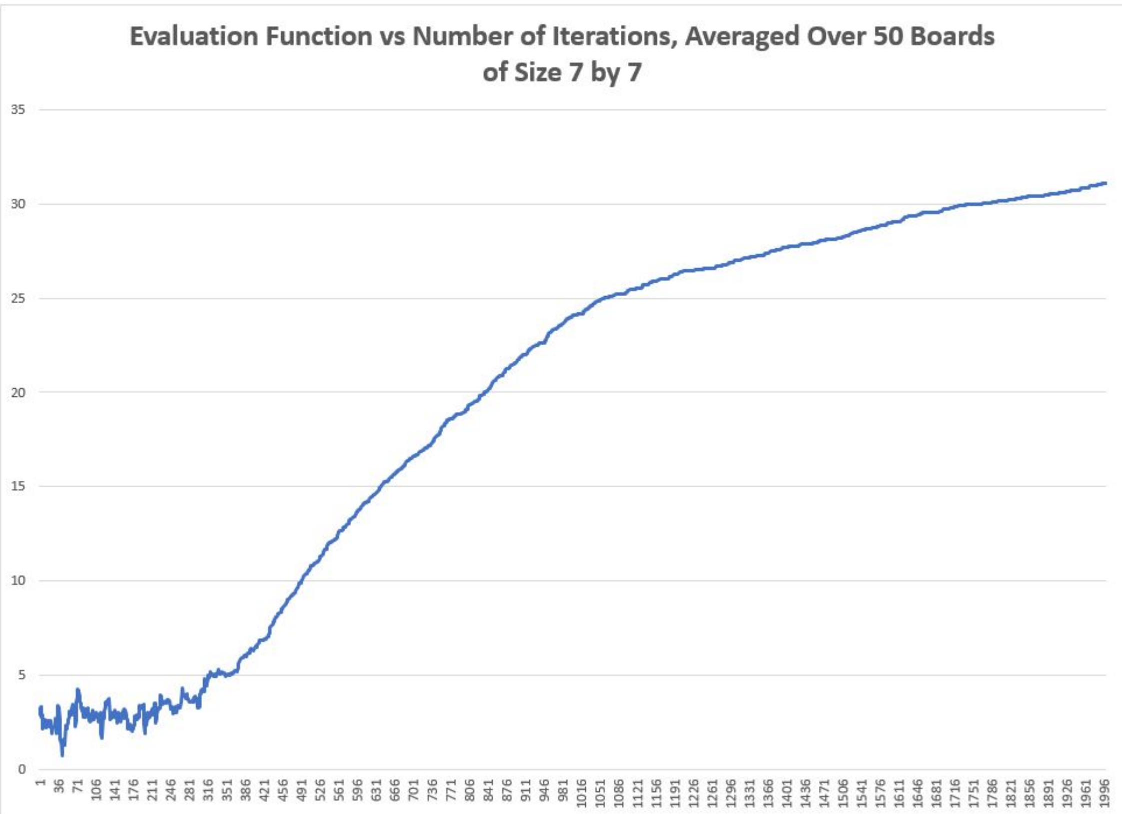
$$T \times d^{\text{iterations}/2} \approx 0 \tag{2}$$

For these constraints, temperature will be high for a significant number of iterations around $iterations/10$, but also near 0 when the iterations are a halfway done, which essentially prevents random walk.

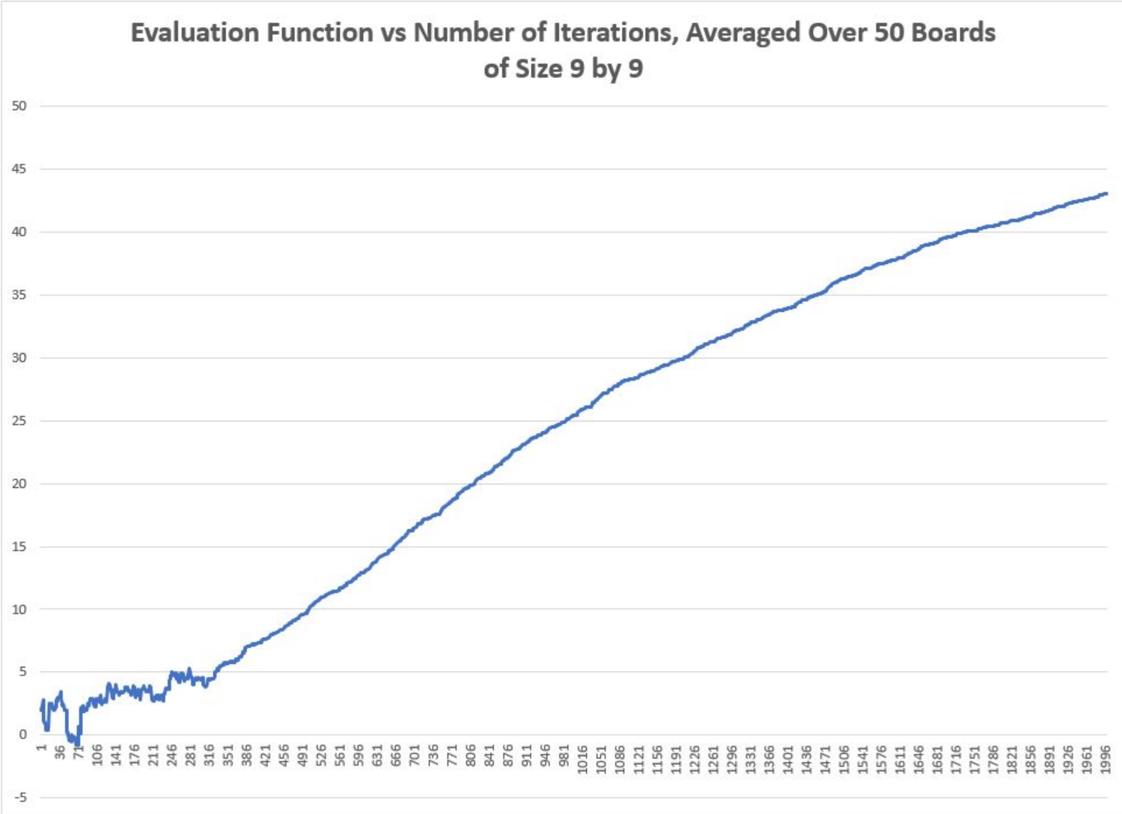
6.1 Size 5 Boards



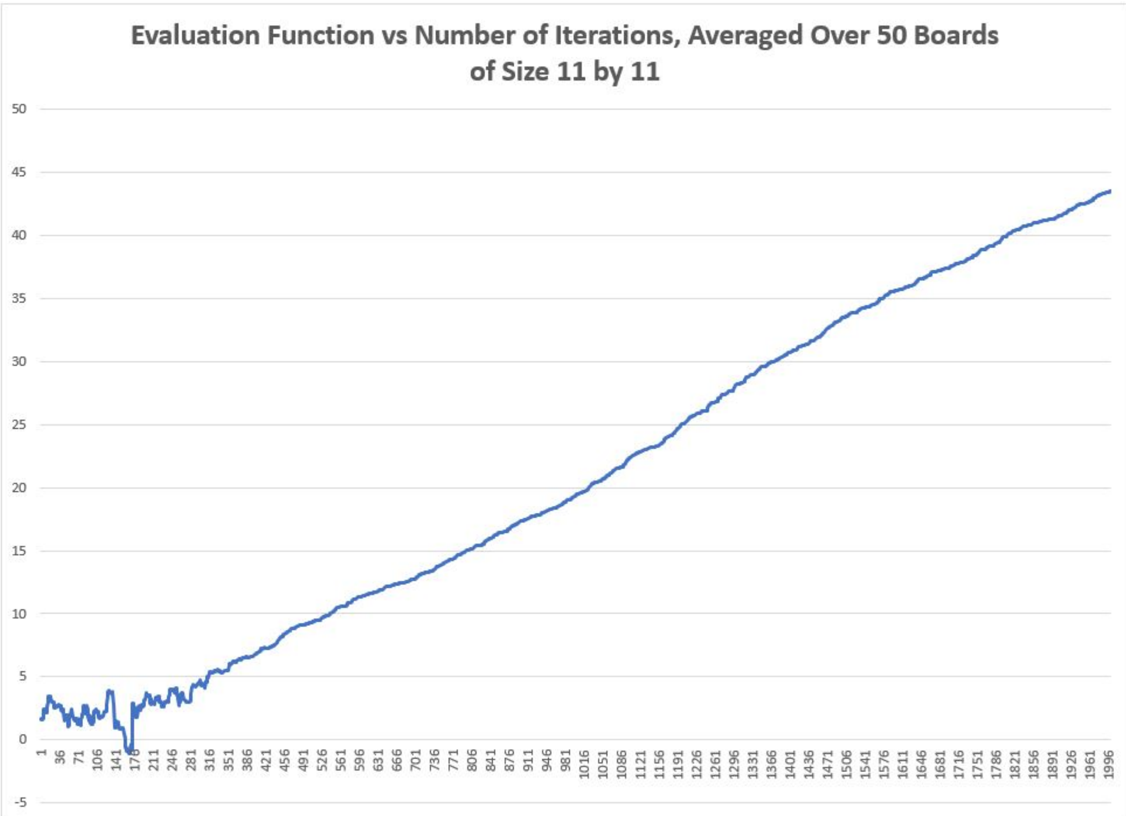
6.2 Size 7 Boards



6.3 Size 9 Boards



6.4 Size 11 Boards



7 Task 7: Propose and Implement a Population-Based Approach

I implemented a fairly standard genetic algorithm, which can be compartmentalized into 7 steps:

1. Initialization of Population
2. Population Evaluation
3. Score Normalization
4. Choosing the Most Fit Individuals
5. Repopulation Through Mating
6. Randomly Mutate Population
7. Check for Stopping Condition

Steps 1 and 2 are not new and have been used in the past Tasks. Score normalization is done through the following algorithm:

1. If the evaluation score for a puzzle is positive, square the number
2. If the evaluation score for a puzzle is negative, set it to 1
3. If the evaluation score for a puzzle is 0, set it to 1
4. Normalize all the numbers by dividing by the total sum

Step 4 involves choosing individuals that live on to the next generation. This is done through a random number generator from 0 to 1; given a randomly generated number x, find the j^{th} index through

$$\sum_{n=0}^{n=j} n^{\text{th}}\text{matrix}_{\text{score}} < x < \sum_{n=0}^{n=j+1} n^{\text{th}}\text{matrix}_{\text{score}} \tag{3}$$

Then comes the mating/repopulation step. The mating algorithm takes pairwise distinct survivor boards and takes the top half from one and the bottom half from the other to combine into another board. This is done until the initial population size is reached.

After, the each board in the resultant population has a chance to be mutated, which means that a cell in the board is changed to another legal number.

7.1 Parameters

This model takes in the following parameters, *side length* of the puzzles, *population size*, *survivor size*, *mutation chance*, and the number of *iterations*. For the graphs, since I had a restricted computing time and this algorithm is more time intensive, I chose relatively small values. Population size = 100, survivor size = 20, and mutation chance = 0.15. The impact of these parameters is also important. By increasing population size, you start off with more "genetic" diversity and you are more likely to evolve into the global maximum of the evaluation function. Diversity similarly applies to the survivor size; if there are too few survivors, then the next population will be "inbred", and not have any variation between them, and if there are too many then the overall genetics of the population changes too slowly. Mutation chance plays a smaller role, but mainly serves to introduce new genetic variation; however, if it is too big, then it causes too many negative mutations, lowering the overall fitness of the population at equilibrium.

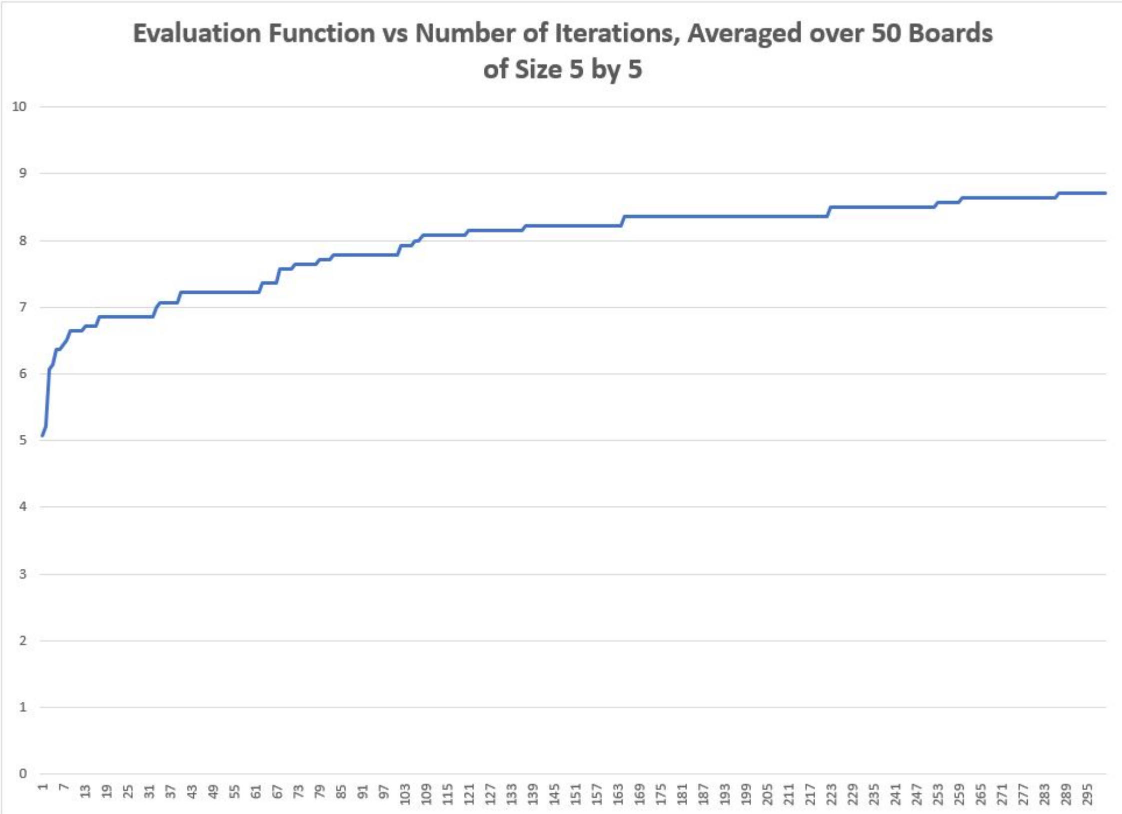
For the graphs, I chose iteration counts such that it would take around the same time as the time of the corresponding size board for the other algorithms.

For small iteration counts, genetic performs much worse than the other algorithms.

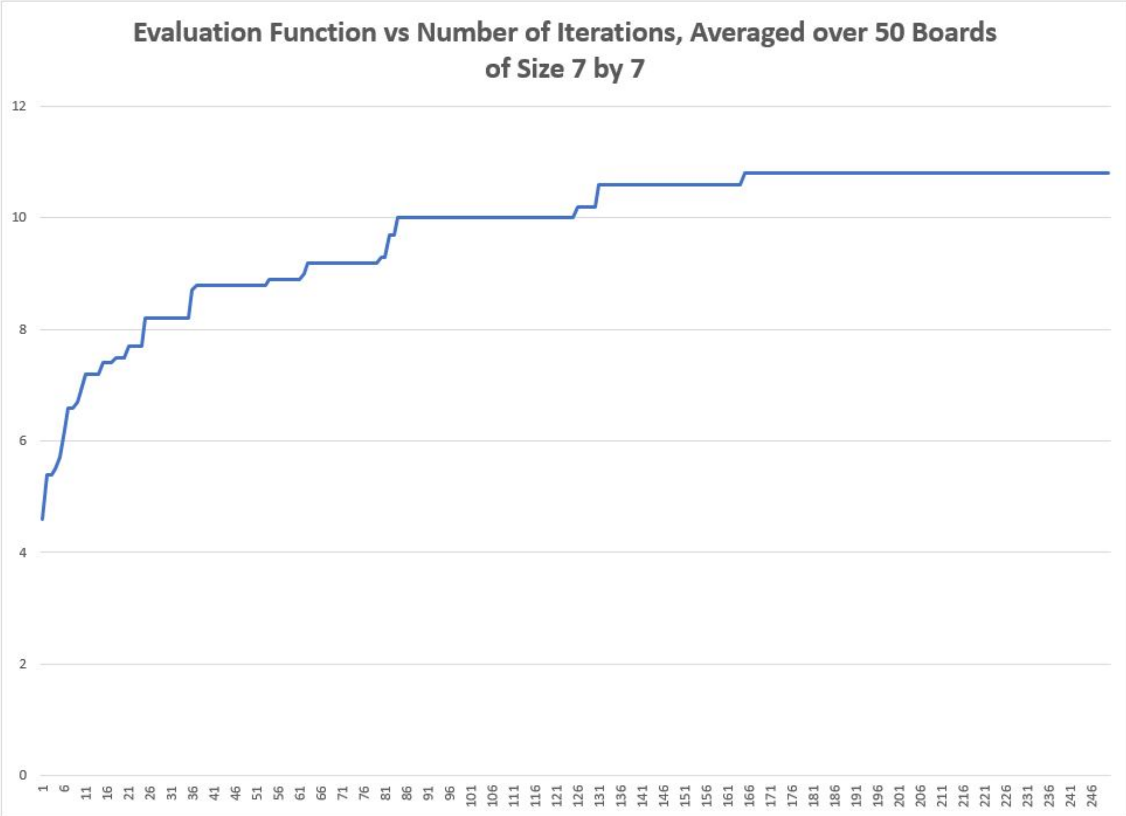
7.2 Note*

Note that because it took a very long time to run my algorithm, for whatever reason, I could only do a small number of iterations for all of the graphs. While this does not affect the simpler algorithms much, it affects the efficacy of this genetic algorithm. Given a much longer runtime, or a faster machine, the evaluation graphs would tend towards an asymptote and reach better values.

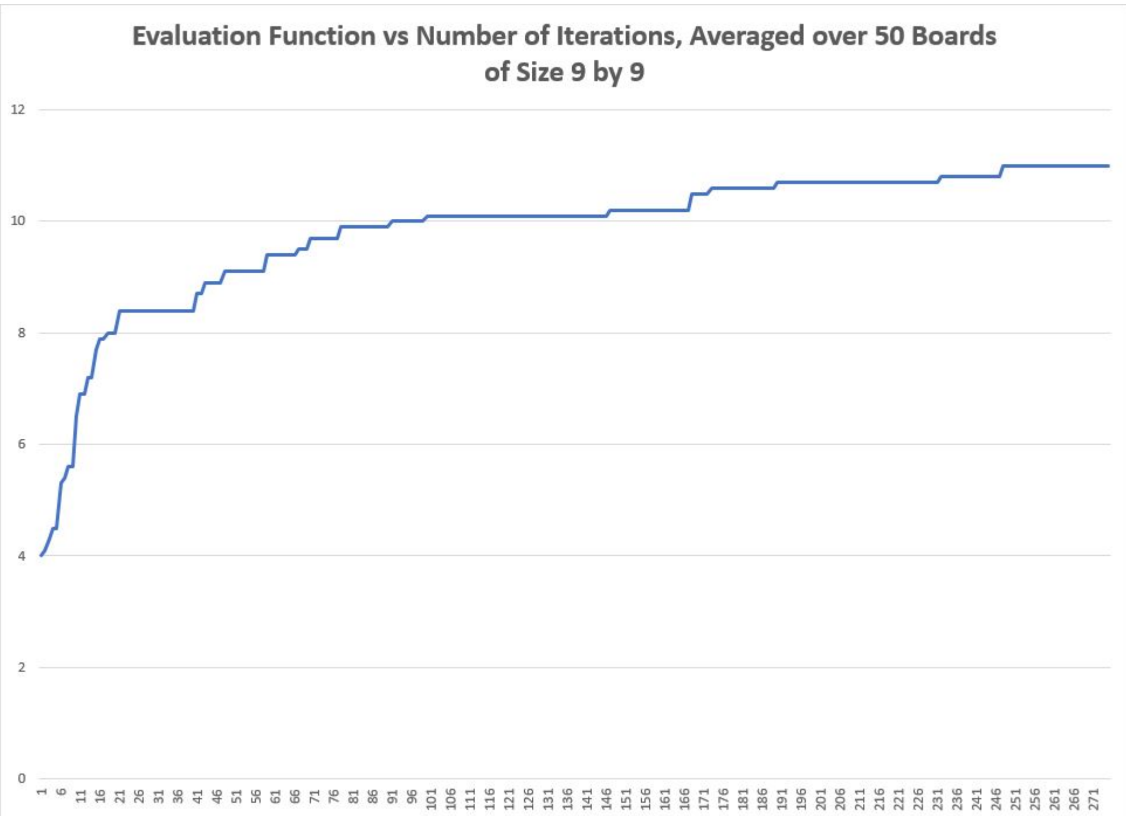
7.3 Size 5 Boards



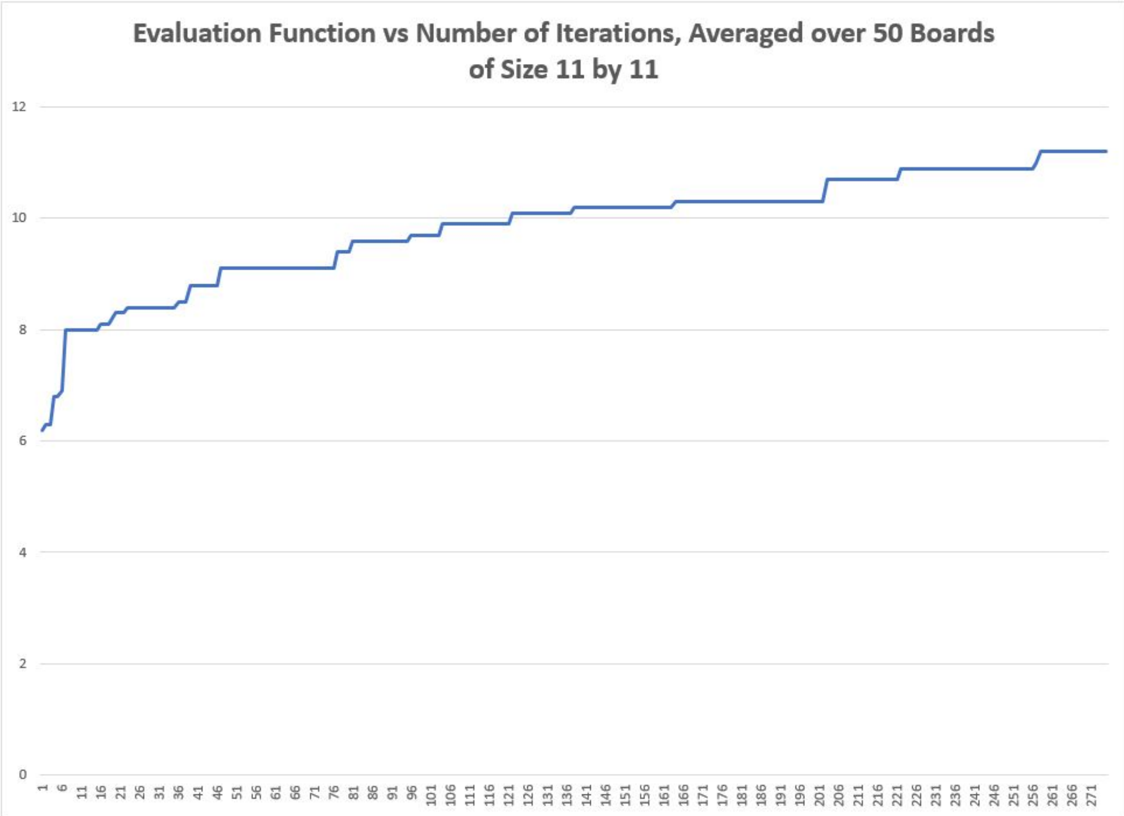
7.4 Size 7 Boards



7.5 Size 9 Boards



7.6 Size 11 Boards



7.7 Genetic Algorithm

As mentioned in section 7, since the laptop the code was run on was slow, a low iteration number for all the algorithms had to be chosen, which affected the genetic algorithm the most. The genetic algorithm has far more steps and its mechanism relies on randomness causing better evaluation scores. Therefore, to see its usefulness requires more evaluation time.