Caleb Howard
s5155126
Trimester 1, 2020

N-Queens C++ Program Report

The N-Queens problem is a famous puzzle in which *N* queens must be placed on on a N*N chessboard where each queen is in a non-attacking position .i.e. No one queen is in the left, right or diagonal path of another queen.
The following report analyses my c++ program that solves the N-Queens problem using a breadth first search (BFS) to find all solutions, and to find one solution, a hill climbing (HC) or simulated annealing (SA) implementation.

Uninformed Search

The uninformed search implemented a Breadth First Search (BFS). The search tree was structured by starting with an empty board then for each new state a queen is placed on an incremental column index [0..N] on the next row. This process repeats until the search is on the Nth row and a board is a valid solution. An example of the search tree is seen is *figure 1*.

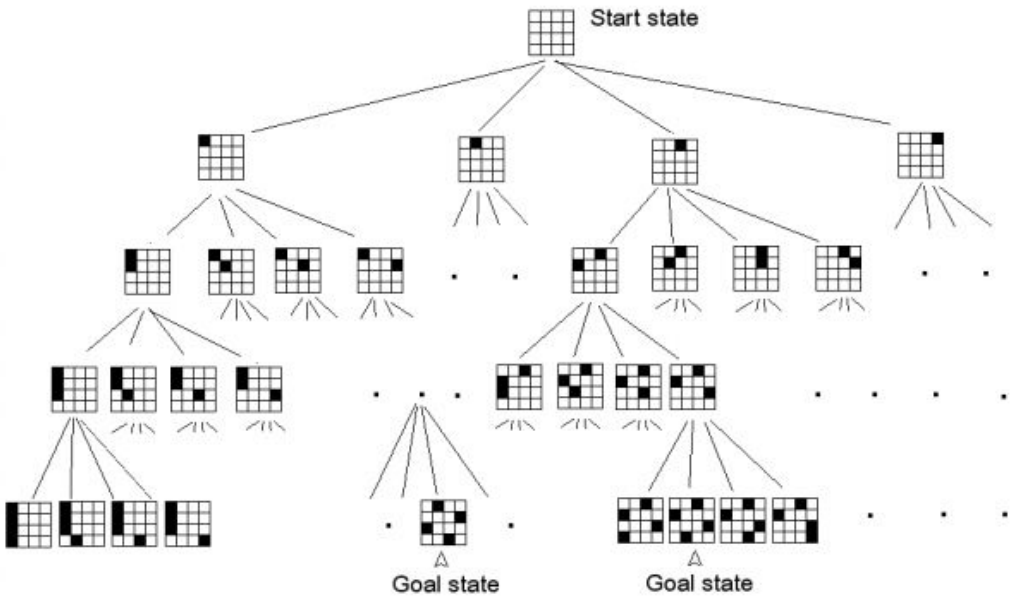


Figure 1. N-Queens BFS tree. (Please note that this image was taken from the Intelligent Systems assessment 1 document and does not belong to me.)

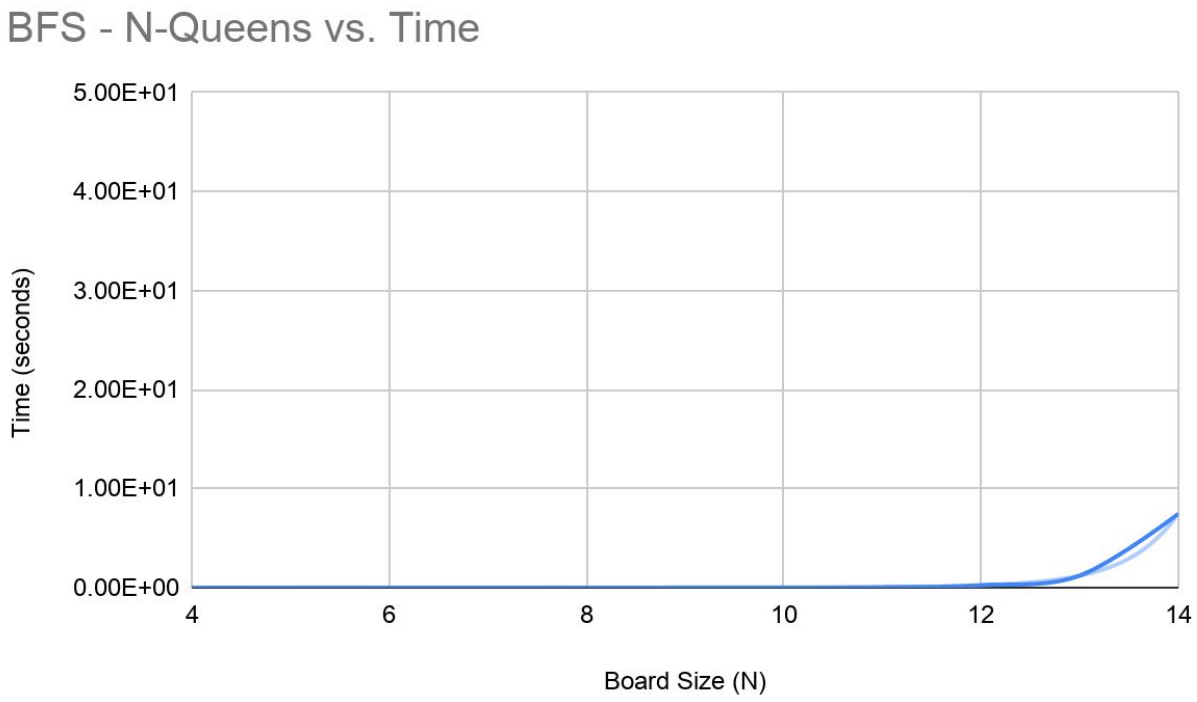
To optimise the BFS and speed up computation an enhancement was made to the code in which only potential states that had a valid placement of the queen were explored further. This enhancement saves memory and unessarcy processing time as less states are stored in the queue and less states are explored, resulting in a dramatic increase in performance, search

time and a higher N value limit, as seen in *figure 4*. Please note that the rest of the document will only refer to BFS as the enhanced BFS search implementation.

Based on the results shown in *figure 4* we can calculate and predict a time for N values that are greater than 20. Below is an example of deducing a time for N=30 by creating a trendline graph formula in google sheets (see *figure 2.*) .

Using the quadratic trendline formula to calculate the predicted time for N=30 we get an estimated 5000 seconds run time.

Figure 2. BFS N-queens vs. Time for N=1 to N=10, excluding N=2 and N=3

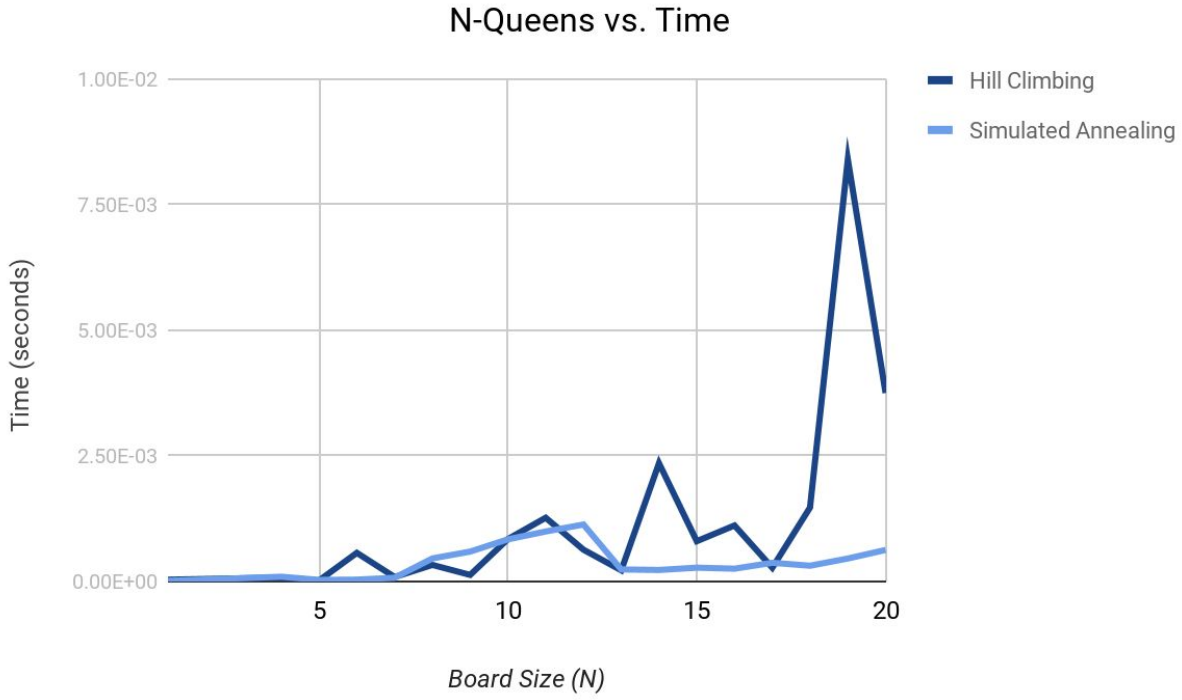


Caleb Howard
s5155126
Trimester 1, 2020

Local Search - Hill Climbing and Simulated Annealing Comparative Study

Two types of local searches were used in this program, a simulated annealing search and a hill climbing search.

Figure 3. N-Queens vs Time for Hill Climbing and Simulated Annealing.



After running the Simulated Annealing algorithm with multiple cooling rates and temperatures, I decided on a value of 4000 for the temperature and a value of 0.99 for the cooling rate. Although different temperature and cooling rate ratios gave better results for different N values, it seems that the 4000, 0.99 ratio worked the overall best on average for N=1 to N=20 for the rates and temperatures I tested.

It can be inferred from *figure 3* that the Hill climbing implementation has an increasing trendline but an unpredictable runtime. The initial random board configuration and the board restarts (where a new board is created if the cost has not changed for a certain amount of queen swaps) will produce boards that may be closer to a solution then those produced on other runs, this can be seen in *figure 3* represented by turning points at N=14 and N=19 where in these cases the random board configurations may have been further from a solution then N=15 and N=20 respectively.

Caleb Howard
s5155126
Trimester 1, 2020

Figure 4. Results For N-Queens Using BFS, HC and SA for N=1 to N=20

N	BFS - All Solutions (seconds)	BFS - All Solutions (After enhancement) (seconds)	HC -One Solution (seconds)	SA - One Solution (seconds)
1	8.2e-04	4.3e-05	3.6e-05	1.2e-05
4	10.6e-04	6.6e-05	7.8e-05	9.5e-05
5	0.002074	0.000184	2e-05	3.1e-05
6	0.004624	0.000565	0.00057	3.7e-05
7	0.118731	0.002432	8.9e-05	7.4e-05
8	1.195625	0.003019	0.00033	0.000459
9	2.004879	0.008679	0.000131	0.000595
10	4.011344	0.041022	0.000844	0.000839
11		0.240534	0.001269	0.000995
12		1.20905	0.000633	0.001134
13		7.45047	0.000223	0.00024
14		47.5205	0.002354	0.000229
15			0.000802	0.000275
16			0.001115	0.000255
17			0.000274	0.000372
18			0.001474	0.000314
19			0.008401	0.000457
20			0.003752	0.000628

Note that there are no solutions for N<1, N=2 and N=3

Simulated Annealing Cumulative Results for N=1 to N=100:
Temp = 4000; Cooling Rate = 0.99: 0.017239 seconds:

Hill Climbing Cumulative Results for N=1 to N=50: 5.321547