Intelligent Systems 2802ICT

N-Queens Problem

Jesse Schneider

2020

# Problem Overview

The point of this report is to report on the findings of using different searching strategies to try and solve the N-Queens problem – both uninformed and informed strategies have been used.

The N-Queens problem is a chess problem – it asks the question: how many unique ways can a number of queens (*n* number of queens) be placed on a chess board that is *n* x *n* in size, where they are unable to attack each other? (i.e. how many ways can 8 queens be placed on an 8 x 8 chess board where they cannot attack each other?). For context – a queen can attack vertically, horizontally or diagonally any number of spaces – so each queen must have a clear horizontal, vertical and diagonals.

# Part A – Uninformed Search

### Algorithm Design – Breadth-First Search

To perform the uninformed search on the N-Queens problem, a Breadth First Search (BFS) algorithm has been implemented. This is a well-known search strategy for uninformed searching.

The ultimate strength of the BFS – the fact that is will always deliver the smallest cost – is lost on this problem, as it is not stopping upon finding a solution; rather it is continuing until all possible solutions have been exhausted. This does not make it any less valid to use though – cost is just simply irrelevant.

### Pseudocode – BFS Search

The below code does not relate to the class is the source code – rather only the BFS implementation:

Number of solutions = 0

add starting board to explored list

create a queue

Add starting board to queue

while queue is not empty:

get next value from the queue

for each column[i] in the board:

if our current row is >= n:

skip

copy our current board

add our new queen to column[i]

keep track of queen coordinates

if this board has already been explored:

skip to next board

create a new board -> pass in current board and its related details

if the current row = n:

check if board == goal state

if goal state:

number of solutions ++

put the board back into the queue for further searching

add the current board to explored list

return the number of solutions

### Results

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N Queens | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Time (seconds) | 0.000 | 0.001 | 0.001 | 0.012 | 0.184 | 3.124 | 69.694 | 1643.712 |

To solve the 8 x 8 board for 8 Queens using the BFS search method takes approximately 27 minutes for my hardware to complete.

Because of this reason I was unable to continue testing due to time constraints on the assignment.