

Illinois Institute of Technology
Department of Computer Science

Homework Assignment 2

CS 480 Artificial Intelligence: Planning and Control
Fall Semester, 2018

1. (20 points) Modify the code to count the total number of assignments attempted during a solution run and print them out.

- a. **How many assignments are made solving 8-queens? 10-queens? 15 queens? 20-queens?**
- b. **Can you solve 100-queens in a reasonable amount of time? If so, how many assignments are made?**

- a. For 8-queens, there are 113 assignments attempted.
For 10- queens, there are 102 assignments attempted.
For 15-queens, there are 1359 assignments attempted.
For 20-queens, there are 199,635 assignments attempted.
- b. No, you cannot solve 100-queens in a reasonable of time.

Since the questions 2,3 asked for column assignment going row by row, this method was used. The code can and has been optimized to look for least constraint row (adding up all the values in a row) and then looking at the least constraint column in that row. I have attached the results of that to the end of this document. The code will be loaded as "optimized_nqueens.py" which can solve 100-queens in about 15 minutes.

2. (40 points) Modify the code to include the least-constraining-value heuristic for determining which order to check new column assignments in.

- a. **How many assignments are made solving 8-queens? 10-queens? 15-queens? 20-queens?**
- b. **Can you now solve 100-queens in a reasonable amount of time? If so, how many assignments are made?**
- c. **What is the largest problem you can solve in less than a minute? How many assignments are made?**

Heuristic:

Given the next row, my algorithm orders the columns in least constraining value using a measure. Thus, the columns are checked in this order. To find the least constraining column, I calculated the impact placing a queen on that spot would have on the rest of the board given a list of placed queens. I counted the spots impacted along the row and column and diagonals. After running assignment tests on the different measures (looking just along the diagonals, looking just along the rows and columns or looking at both), looking just along the diagonals and looking at all of them yielded the best assignments (tied). Looking at just the

diagonals yielded the best time, so I used this as my measure. The following is a visualization of the method in a board size 4:

Along rows and columns and diagonals no queens:

1	1	1	1
1	1		
1		1	
1			1

10			

1	1	1	1
1	1	1	
	1		1
	1		

10	10	10	10

Starting Block
Row 0 counting
For column 0

Now (0,0) = 7

Now for (0,1)

Continuing for row 0

This orders the points (10,0,0), (10,0,1), (10,0,2), (10,0,3) and (0,0) gets chosen since it is first.

Along rows and columns and diagonals after a queen placed:

Q	-	-	-
-	-	1	1
-	1	-	1
-		1	-

Q	-	-	-
-	-	5	
-		-	
-			-

Q	-	-	-
-	-	1	1
-		-	1
-	1		-

Q	-	-	-
-	-	5	4
-		-	
-			-

Starting Block
Row 1 counting
For column 2

Now (1,2) = 7

Now for (1,3)

Continuing for row 1

This orders the points (4,1,3), (5,1,2) and (1,3) gets chosen since it is first

NOTE: Only **along diagonals** was used since it yielded the best assignments and fastest time. Rows, columns and diagonals (above counting) yielded the same amount of assignments with slower times

- For 8-queens, there are 62 assignments attempted in 0.009 seconds.
For 10- queens, there are 10 assignments attempted in 0.003 seconds.
For 15-queens, there are 149 assignments attempted in 0.062 seconds.
For 20-queens, there are 5752 assignments attempted in 3.608 seconds.
Note: My machine is slow, so times will drastically vary
- No, you cannot solve 100-queens in a reasonable of time where reasonable time is at least 15 minutes).
- The largest problem solved under a minute is 29 with 33108 assignments attempted in 45.866 seconds.

- 3. (40 points) Incorporate arc-consistency checking into the program. (You will need an extra data structure to remember the allowable values for each row.)**
- How many assignments are made solving 8-queens? 10-queens? 15-queens? 20-queens?**
 - Can you now solve 100-queens in a reasonable amount of time? If so, how many assignments are made?**
 - What is the largest problem you can solve in less than a minute? How many assignments are made?**

My implementation runs arc consistency on the least constraint value. When placing a new queen, the algorithm checks if it is consistent (this does not count as an assignment since if it is not consistent, we will backtrack). If the arc consistency list contains empty lists, then it is inconsistent. A bit map is used to track queens and update the consistency map. Here are the results:

- For 8-queens, there are 33 assignments attempted in 0.0270 seconds.
For 10- queens, there are 10 assignments attempted in 0.006 seconds.
For 15-queens, there are 70 assignments attempted in 0.152 seconds.
For 20-queens, there are 2199 assignments attempted in 8.559 seconds.
 - No, you cannot solve 100-queens in a reasonable of time where reasonable time is at least 15 minutes).
 - The largest problem solved under a minute is 27 with 3544 assignments attempted in 25.961 seconds. The assignments have dropped drastically but time has increased (solves 29 from before in 78.360 seconds with 9327 < 33108 assignments attempted) since there is overhead to update the bitmap and then the consistency list.
- 4. (20 points extra credit) Implement min-conflicts local search.**
- How does the time to solve 8, 10, 15, 20, 100 (or more) queens compare with the backtracking search?**
 - What is the largest problem you can solve in less than a minute?**
- For 8-queens, there are 20 assignments attempted (irrelevant since randomly assigned initially) in 0.0155 seconds.
For 10- queens, there are 126 assignments attempted in 0.044seconds.
For 15-queens, there are 47 assignments attempted in 0.040 seconds.
For 20-queens, there are 73 assignments attempted in 0.068 seconds.
For 100-queens, there are 101 assignments attempted in 2.298 seconds.
For 200-queens, there are 168 assignments attempted in 13.340 seconds.
For 300-queens, there are 250 assignments attempted in 45.6614 seconds.
 - This question is relative since the initial setting of the board is done randomly. Starting the queens on 0th row and finding the largest conflicting queen and then looks for a min conflict would be a good optimization to have strict results.

I ran the algorithm incrementing board size each time to find the size that would make it run over a minute. “the largest problem” it solved in that instance was 325 with 278 assignments in 54.722 seconds.

Below are the results after optimizing the question by not going row by row iteratively but by choosing the least constraint row and then the least constraint column:

- 2. (40 points) Modify the code to include the least-constraining-value heuristic for determining which order to check new column assignments in.**
- How many assignments are made solving 8-queens? 10-queens? 15-queens? 20-queens?**
 - Can you now solve 100-queens in a reasonable amount of time? If so, how many assignments are made?**
 - What is the largest problem you can solve in less than a minute? How many assignments are made?**

My implementation determined the order of the columns and the points in those columns by finding the least constraining column depending if a queen was placed and previous queens placed. Then it finds the least constraining points in that column with the same rules. The constraining factor is measuring the impact a new queen will have on the board (how many blocks impacted). A series of test were run to vary the block definitions. The first just looks at the row and column block respective of the point chosen in the least constraining column. The second looks at the only the diagonal blocks respective of the point chosen in the least constraining column. Finally, the last one uses both rows and columns and the diagonals impacted. These are the results:

a.

	8 Queens		10 Queens		15 Queens		20 Queens	
Blocks Impacted	Assignments	Time (sec)	Assignments	Time (sec)	Assignments	Time (sec)	Assignments	Time (sec)
Horizontal and Vertical	69	0.061	34	0.072	80	0.357	27	0.507
Diagonals	46	0.027	12	0.023	55	0.197	150	1.080
Both	39	0.048	10	0.052	15	0.242	746	9.641

As seen just checking for the least constraining block based on the columns and rows are slow at first and becomes faster as board size increases. Furthermore, the use of both methods results in faster time for board size less than 20. On average, looking at the diagonals gives the best assignments since it stays between the other two options. **Note: My machine is slow, so times will drastically vary**

- b. No, 100 cannot be run less than 5 minutes (reasonable time definition) for any of the versions. **Note: My machine is slow, so times will drastically vary**

c.

Blocks Impacted	Assignments	Time (sec)	Maximum Blocks
Horizontal and Vertical	73	56.863	55
Diagonals	1796	22.888	27
Both	765	58.166	37

Note: My machine is slow, so times will drastically vary

3. (40 points) Incorporate arc-consistency checking into the program. (You will need an extra data structure to remember the allowable values for each row.)
- How many assignments are made solving 8-queens? 10-queens? 15-queens? 20-queens?
 - Can you now solve 100-queens in a reasonable amount of time? If so, how many assignments are made?
 - What is the largest problem you can solve in less than a minute? How many assignments are made?

My implementation runs arc consistency on the preciously discussed methods of least constraint value. When placing a new queen, the algorithm checks if it is consistent (this does not count as an assignment since if it is not consistent, we will backtrack). If the arc consistency list contains empty lists, then it is inconsistent. Here are the results:

a.

As seen, the assignment rate as decreased but impacted the time. This is due to the

	8 Queens		10 Queens		15 Queens		20 Queens	
Blocks Impacted	Assignments	Time (sec)	Assignments	Time (sec)	Assignments	Time (sec)	Assignments	Time (sec)
Horizontal and Vertical	46	0.077	29	0.074	51	0.321	24	0.581
Diagonals	32	0.061	12	0.027	47	0.220	112	1.363
Both	29	0.082	10	0.048	15	0.228	436	9.05

overhead of maintaining and use an extra data structure. **Note: My machine is slow, so times will drastically vary**

- b. 100-queens was solvable in 998.024 seconds with 106 assignments using the horizontal and vertical block impact. This is about 17 minutes, but I don't know if

it is a reasonable timeframe. Note: My machine is slow, so times will drastically vary

c.

Blocks Impacted	Assignments	Time (sec)	Maximum Blocks
Horizontal and Vertical	67	53.813	55
Diagonals	1796	18.793	27
Both	73	54.218	48

Note: My machine is slow, so times will drastically vary