Vectorized Method for Solving the n-queens Problem using Bohrium

Larsen, Mads Ohm

Publication date:
2016

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (APA):
Vectorized Method for Solving the $n$-queens Problem using Bohrium

Mads Ohm Larsen <ohm@nbi.ku.dk>

Introduction
On a normal $8 \times 8$-chessboard we find that 8 queens can be positioned in

$$\binom{64}{8} = \frac{64!}{8! \cdot (64-8)!} = 4,132,851,600$$

different legal ways. This is too many to brute force, especially as we go for larger $n$.

For the $8 \times 8$-queens problem we only have 92 distinct solutions.

Backtracking Algorithm
Since only one queen are allowed in each row and column, we can skip placing another queen in the same column or row in the backtracking algorithm.

The problem is simple, but non-trivial to solve. Using the knowledge stated before, we only have to check

$$8^8 = 2^24 = 16,777,216$$

placements, instead of the roughly four and a half billion legal placements.

Using Permutations
Instead of looking for the right combinations, we can instead compute permutations of rows, with a queen in each col-

umn. There are

$$8! = 40,320$$

permutations of a $8 \times 8$-chessboard’s rows.

Because we construct the rows with a queen in each column, we now know we only have to check the diagonals, to see if the current permutation is a solution.

```python
import numpy as np
# import bohrium as np
from itertools import permutations
def nqueens(n):
    # Generate all permutations of an identity boards
    boards = list(permutations(np.arange(8)))
    # Attack the flipped boards as another dimension
    rotation_boards = np.array((boards, np.flip(boards)))
    # Calculate all the traces, from -n to n for all the boards
    n = np.max([
        np.trace(rotation_boards,
            axis1=0,
            axis2=1)
    for i in range(-n, n)])
    # Count the number of Is in the maximum of the traces
    print(np.sum(n == 1),  "solutions for", n, "by", n, "board ", nqueens(8))
```

```python
def nqueens(8) # => 92 solutions for 8 by 8 board.
```