

N-Queens Problem Using Hill Climbing Approach

Algorithm :

Step 1 : Initialize

- Generate a random initial state (current-state) where each queen is in a random column of its row
- Calculate the number of attacking pairs current-cost for current-state.

Step 2 : Repeat until solution found

- Generate all neighbors of current-state:
 - * For each row, move the queen to every other column in that row
 - * Each move creates a neighbor state
- Calculate the cost (attacking pairs) for each neighbor
- Select the neighbor with the lowest cost (best-neighbor)

Step 3 : Move or stop

- If best-neighbor cost < current-cost:
 - * Set current-state = best-neighbor
 - * Set current-cost = best-neighbor cost
- Else (no improvement):
 - * Stop (local maximum reached or solution found)

Step 4 : Check solution

- If current-cost == 0 \rightarrow solution found
- Else \rightarrow may start with a new random state

Step 5 : Return

current-state as the solution

Output:

Initial state : [3, 1, 0, 0]

			Q
	Q		
Q			

Attacking pairs = 2

Neighbours and their conflicts

(1, 3, 0, 0) \rightarrow Attacking pairs : 1

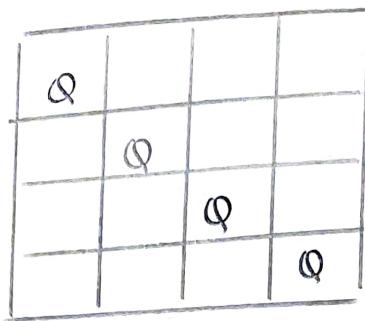
	Q		
			Q
		Q	
Q			

Attacking pairs = 2

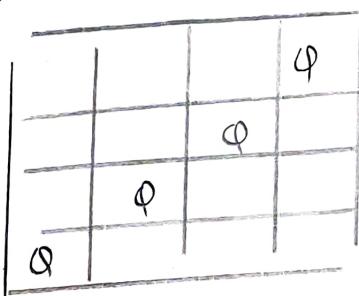
\rightarrow (0, 1, 3, 0) \rightarrow Attacking pairs : 1

			Q
		Q	
Q			

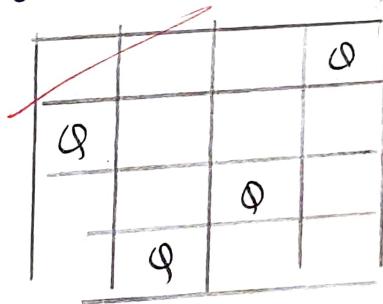
$\rightarrow (0, 1, 2, 3) \rightarrow$ Attacking pairs: 6



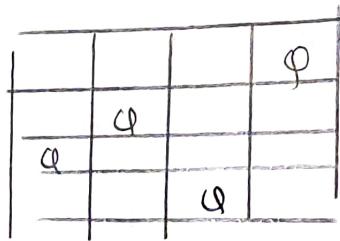
$\rightarrow (3, 2, 1, 0) \rightarrow$ Attacking pairs: 6



$\rightarrow (3, 0, 2, 1) \rightarrow$ Attacking pairs: 1



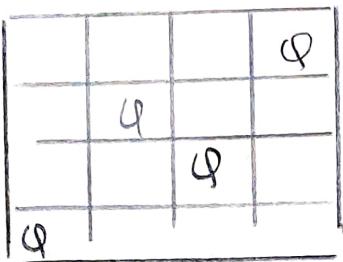
$\rightarrow (3, 1, 0, 2) \rightarrow$ Attacking pairs: 1



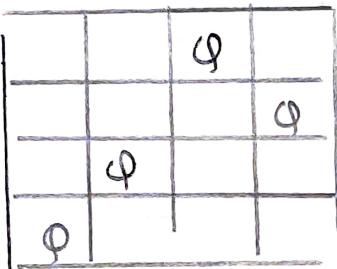
Chosen state $[1, 3, 2, 0]$ attacking pairs: 1

Step 2: Current state: $[1, 3, 2, 0]$ attacking pairs
Neigh. bows and their cats:

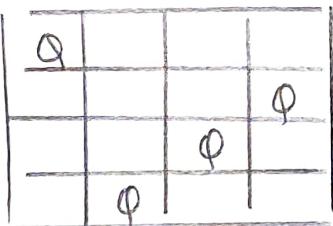
$\rightarrow (3, 1, 0, 0) \rightarrow$ Attacking pairs : 2



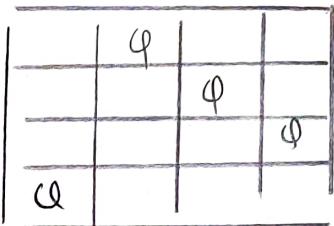
$\rightarrow (2, 3, 1, 0) \rightarrow$ Attacking pairs : 2



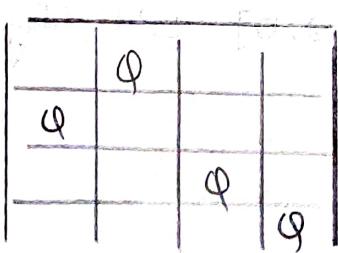
$\rightarrow (0, 3, 2, 1) \rightarrow$ Attacking pairs : 4



$\rightarrow (1, 2, 3, 0) \rightarrow$ Attacking pairs : 4



$\rightarrow (1, 0, 2, 3) \rightarrow$ Attacking pairs : 2



$\rightarrow [1, 3, 0, 2] \rightarrow$ Attacking pairs: 0

	Q		
			Q
Q			
		Q	

Chosen state: $[1, 3, 0, 2]$

Final solution: $[1, 3, 0, 2]$

~~Brute force~~
~~2^n / n!~~
N-Queens Problem Using Simulated Annealing

Algorithm:

1) Initialize

- Generate a random current-state (queen positions)
- Set temperature $T = \text{initial_temperature}$.

2) Repeat while $T > \text{min_temperature}$.

- Pick a random neighbor by moving one queen
- calculate its attacking pairs (neighbor-cost)
- If neighbor is better accept it.
- If worse accept it with probability $\exp(-(neighbor_cost - current_cost)/T)$.

• Reduce temperature: $T = T * \alpha$.

• Stop if attacking pairs = 0 (solution found)

3) Return the final state (solution or best found)

Output: [0, 0, 3, 2] Attacking pairs: 3

Initial state: [0, 0, 3, 2] Attacking pairs: 3

Q			
Q			
		Q	
	Q		

Step 1: Current state: [0, 0, 2, 2] Attacking pairs: 4

Temperature: 100.00

Q			
Q			
		Q	
	Q		

Step 2: Current state: [0, 0, 2, 1] Attacking pairs: 3,

Temperature: 95.00

Q			
Q			
		Q	
	Q		

Step 3: Current state: [2, 0, 2, 1], attacking pairs 2.

Temperature 90.25

		Q	
Q			
		Q	
	Q		

Step 4: Current State: [2, 0, 3, 1] Attacking pairs: 0
Temperature: 85.74

			Q	
		Q		
			Q	
	Q			

Final state: [2, 0, 3, 1], Attacking pairs: 0

~~5th
22/9/20~~