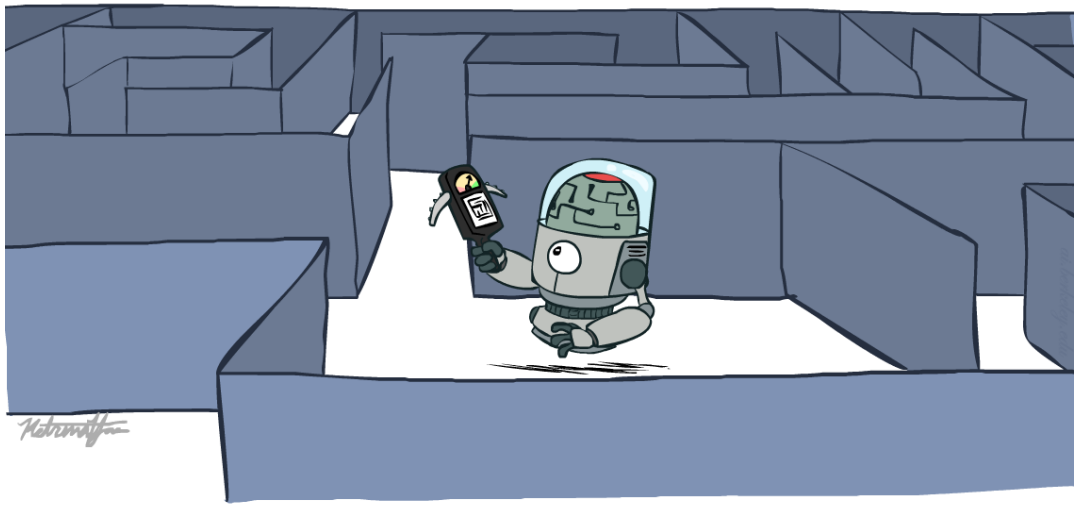
Non-Classical Search



Georges Sakr
ESIB

Today

- Hill Climbing
- Genetic Algorithm

Iterative improvement algorithms

- In many problems, path is irrelevant;
- The goal itself is the solution

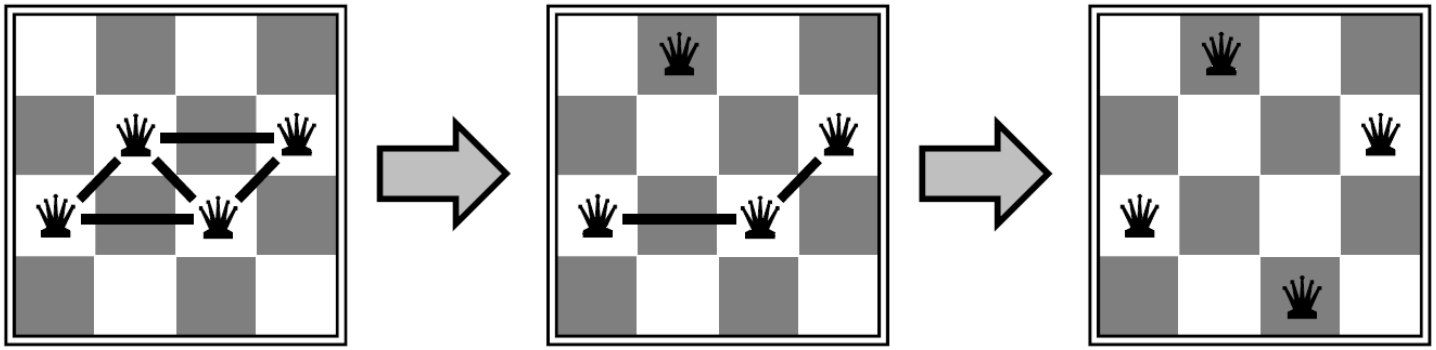
- Example:
  - Find optimal time schedule for all classes
  - Map coloring problem
  - TSP

- In such cases, can use iterative improvement algorithms; keep a single state ("current") and try to improve it

- Constant space, suitable for online and offline learning

Example: n-queens

- Put n queens on an n x n board with no 2 queens attacking each other (same row, column and diagonal)
- Strategy: Move a queen to reduce the number of conflicts



- Almost always solve n-queens problems almost instantly for very large n: >1,000,000

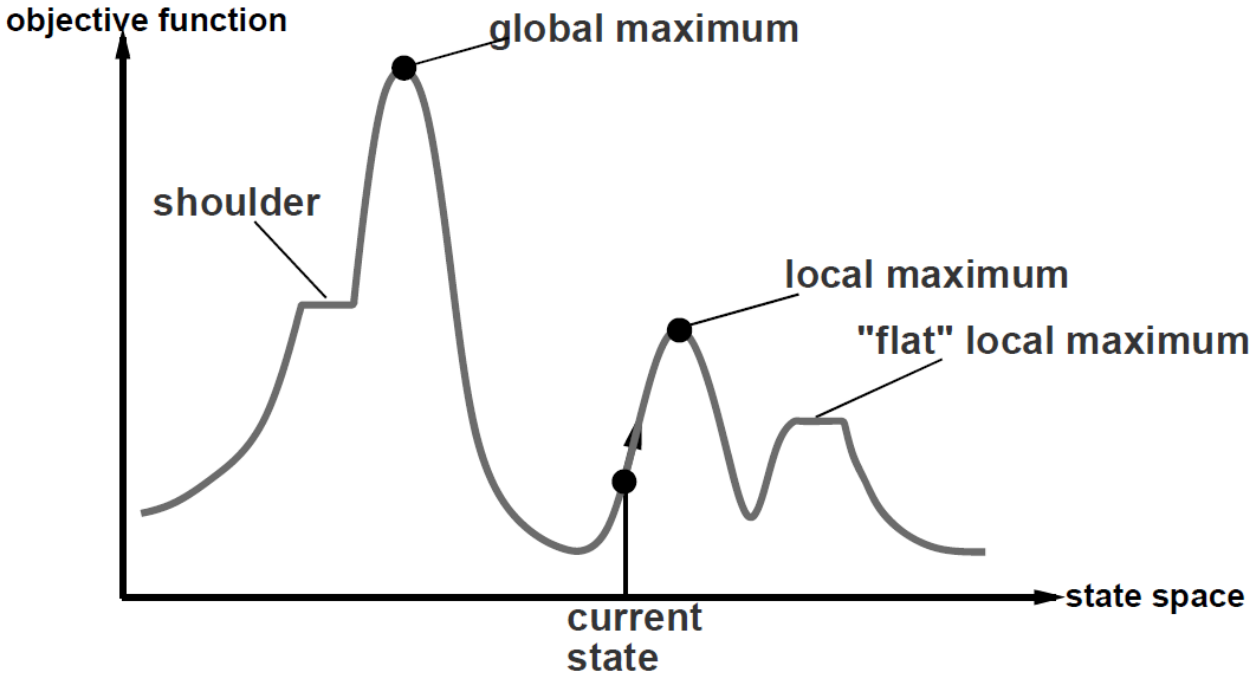
Hill Climbing (Gradient descent/ascent)

```
function HILL-CLIMBING(problem) returns a state that is a local maximum
    inputs: problem, a problem
    local variables: current, a node
                     neighbor, a node

    current ← MAKE-NODE(INITIAL-STATE[problem])
    loop do
        neighbor ← a highest-valued successor of current
        if VALUE[neighbor] ≤ VALUE[current] then return STATE[current]
        current ← neighbor
    end
```

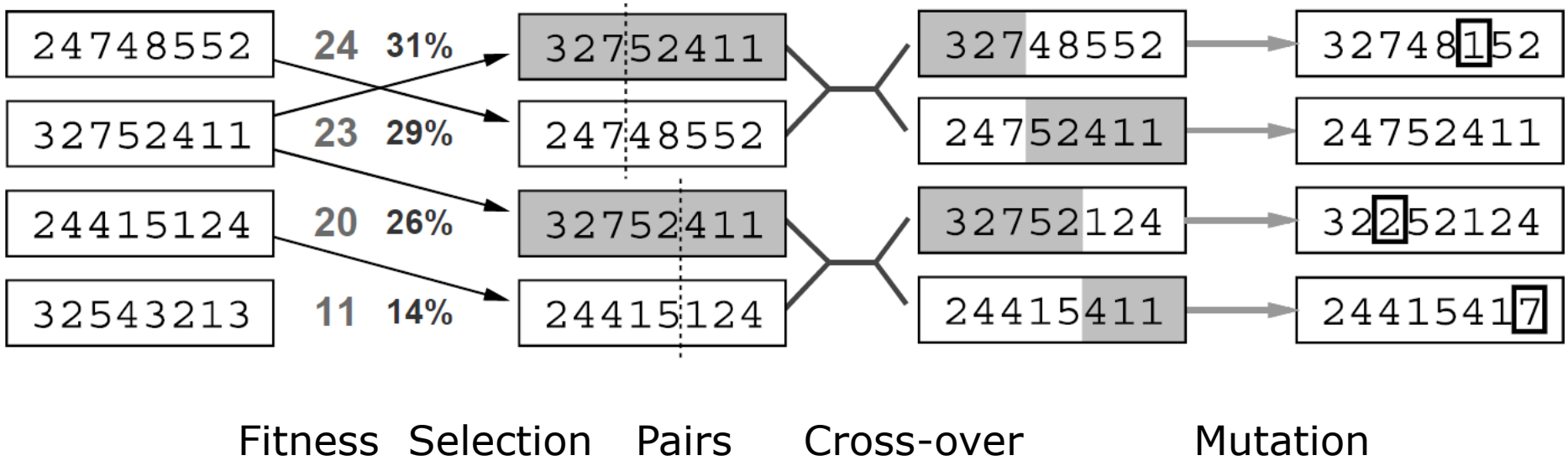
Hill Climbing - Continued

- Consider state space landscape



- Random-Restart hill climbing overcomes local maxima --- Trivially Complete
- Random sideways moves escapes from shoulder --- loops on flat maxima

Genetic Algorithm



Idx=np.random.choice(range(N), 1, p=fitness)[0]
Newpopu.append(population[idx])
C1[0:x+1]+C2[x+1:n]

GA requires states to be encoded as strings

- Crossover helps if substrings represents meaningful states.