Artificial Intelligence

Muhammad Bilal Shahnawaz

bilal.shahnawaz@numl.edu.pk

Outline

- Last Class
  - Informed Search Algorithms
    - Heuristics function
    - Best first search algorithm
    - Greedy Best First search
    - A* search
- Today
  - Beyond Classical Search
  - Types of Problems
  - Optimization problems
  - Local Search Algorithms
    - Hill Climbing
    - Hill climbing features
    - Hill climbing Problems
    - Hill Climbing variants

Types of Problems

- Planning Problems
  - We want a path to a solution
  - Usually want an optimal path
  - Incremental formulations

- Identification Problems
  - We actually just want to know what the goal is
  - Usually want an optimal goal
  - Complete-state formulation
  - Iterative improvements algorithms

Search Algorithms So Far

- Designed to explore search space systematically:
  - keep one or more paths in memory
  - record which have been explored and which have not
  - When a goal is found, a path to goal represents the solution to the problem
  - Example:
    - The solution to the traveling in Romania problem is a sequence of cities to get to Bucharest

Contd..

- If the path to the goal does not matter, we might consider a different class of algorithms that do not worry about paths at all.
  - Local search algorithms

Local search algorithms

- In many optimization problems, the path to the goal is irrelevant; the goal state itself is the solution
- Then, State space = set of "complete" configurations
- Algorithm goal:
  - Find **optimal** configuration (e.g. traveling salesman problem(TSP))
  - Find configuration satisfying constraints, e.g., **n-queens**

- In such cases, we can use local search algorithms

Contd..

- Local search algorithms operate using *a single current node* and generally move only to neighbors of that node.
- Ease up on **completeness and optimality** in the interest of **improving time and space complexity**
- Although local search algorithms are not systematic, they have two key advantages:
  - They use very little memory (usually a constant amount)
  - They can often find reasonable solutions in large or infinite (continuous) state spaces.

Contd..

- In addition to finding goals, local search algorithms are useful for solving pure **optimization problems**, in which **the aim is to find the best state according to an objective function**.
  - In optimization problems, **the path to goal is irrelevant** and the goal state itself is the solution.
  - In some optimization problems, **the goal is not known and the aim is to find the best state**.

- Example: Hill Climbing Algorithm

Hill Climbing Algorithm

- A local search algorithm which continuously moves in the direction of increasing elevation/value to find the peak of the mountain or best solution to the problem.
- moves to the best successor of the current node according to an objective function.
  - It moves in direction of uphill (hill climbing).
  - terminates when it reaches a peak value where no neighbor has a higher value.
  - No need to maintain and handle the search tree or graph as it only keeps a single current state and the value of the objective function.
  - Mostly used when a good heuristic is available.
- Used for optimizing the mathematical problems. E.g. Traveling-salesman Problem

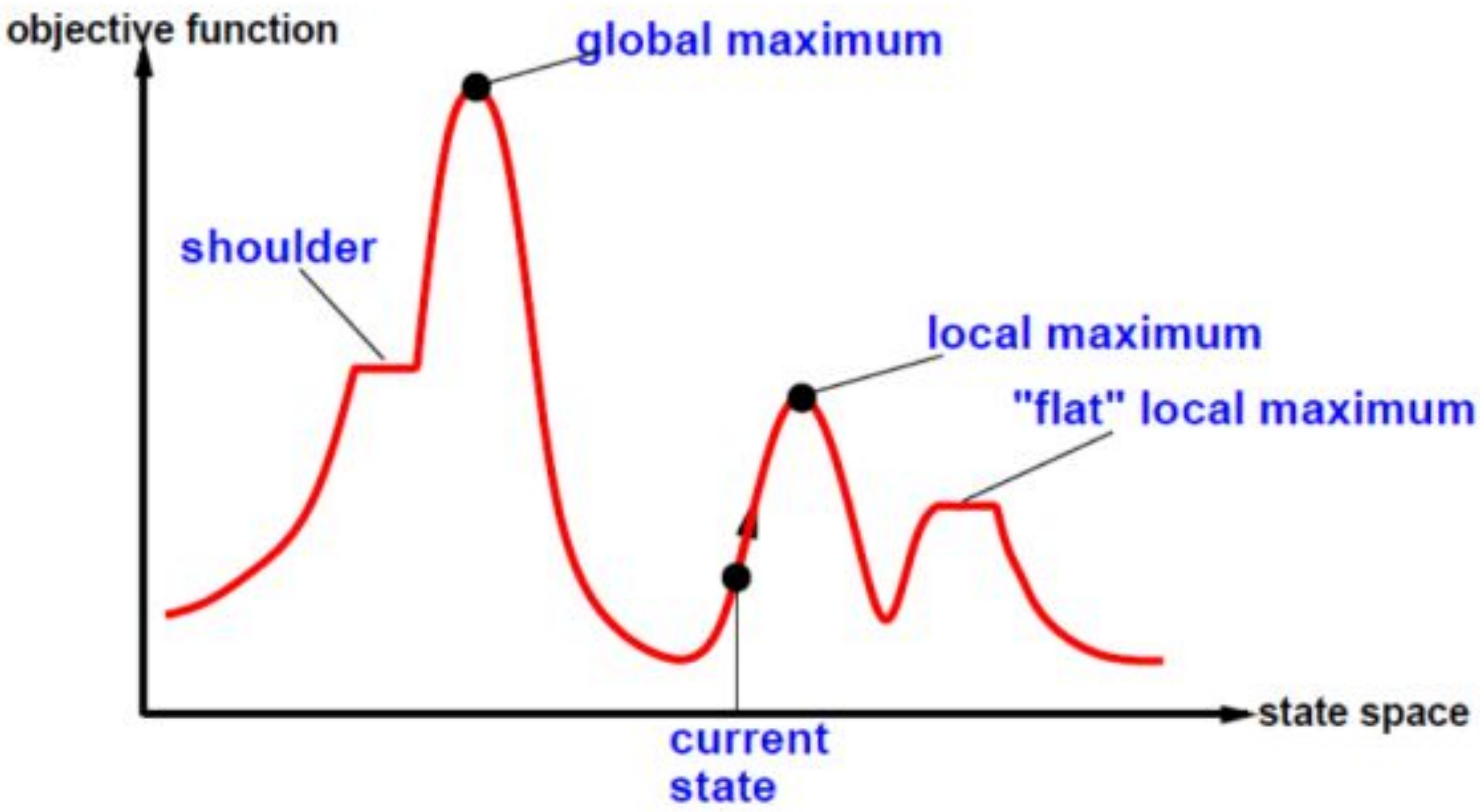
Features of Hill Climbing

- **Local Search Algorithm**
  - Only knows about the local domain, no knowledge of global domain/problem
- **Greedy approach**
  - Search moves in the direction which optimizes the cost.
  - Continues until finds the best move and stops when no best move found
  - grabs a good neighbor state without thinking ahead about where to go next
- **No backtracking:**
  - It does not backtrack the search space, as it does not remember the previous states. Just terminates

State-space Diagram

- To understand local search, it is useful to consider the **state-space landscape**.
  - A graphical representation of the hill-climbing algorithm which is showing a graph between various states of algorithm and Objective/cost function
- The aim is to find the highest peak - **a global maximum** or lowest peak- a global minima
- Hill-climbing search modifies the current state to try to improve it.

Contd..



objective function
global maximum
shoulder
local maximum
"flat" local maximum
state space
current
state

Different regions in the state space

- **Local Maximum**
  - Local maximum is a state which is better than its neighbor states, but there is also another state which is higher than it.
- **Global Maximum**
  - Global maximum is the best possible state of state space landscape. It has the highest value of objective function.
- **Current state**
  - It is a state in a landscape diagram where an agent is currently present.
- **Plateau**
  - A flat area of the state-space landscape. It can be:
  - **Flat local maximum**
    - All the neighbor states of current states have the same value and no uphill exit exists
  - **Shoulder**
    - A flat area from which has an uphill edge and progress is possible

Algorithm

- **Step 1:** Evaluate the initial state, if it is goal state then return success and Stop.
- **Step 2:** Loop Until a solution is found or there is no new operator left to apply.
- **Step 3:** Select and apply an operator to the current state.
- **Step 4:** Check new state:
  - If it is goal state, then return success and quit.
  - Else if it is better than the current state then assign new state as a current state.
  - Else if not better than the current state, then return to step2.
- **Step 5:** Exit.

Hill Climbing Example: n-queens

- Put n queens on an n × n board with no two queens on the same row, column, or diagonal

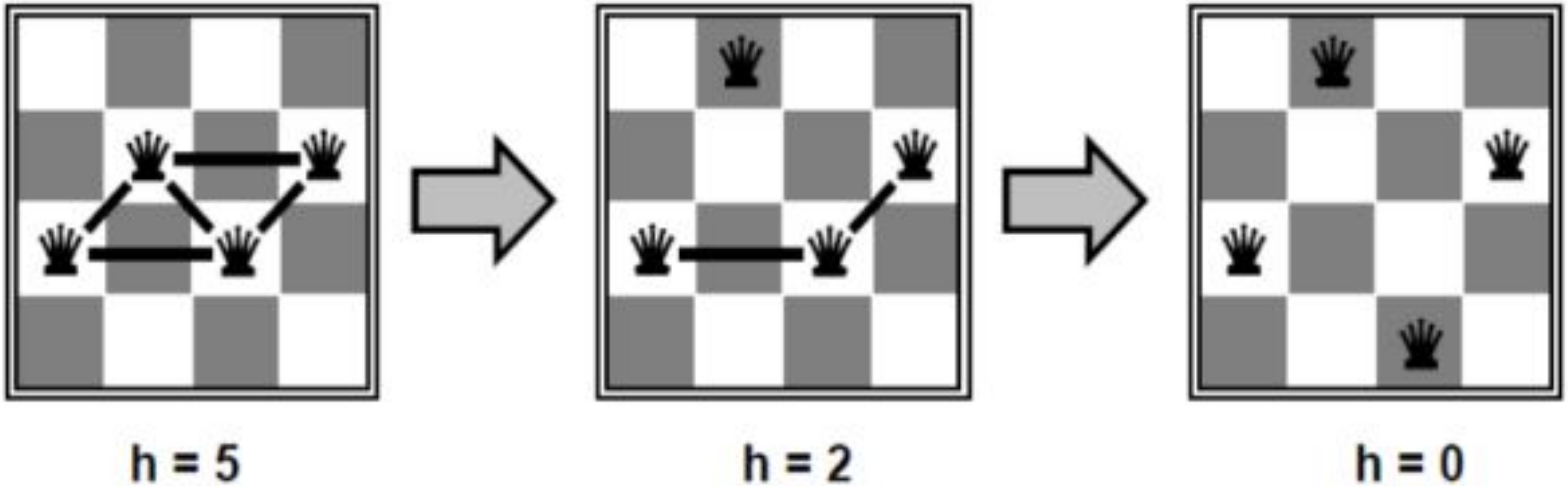
h = 5

  - Here, goal is initially unknown but is specified by constraints that it must satisfy

Contd..

- Move a queen to reduce number of conflicts.
  - **Objective function**: number of conflicts (no conflicts is global minimum)
  - **The successors (successor function)** of a state are all possible states generated by moving a single queen to another square in the same column
  - **The heuristic cost function h** is the number of pairs of queens that are attacking each other, either directly or indirectly.
  - **The global minimum of this function is zero**, which occurs only at perfect solutions.

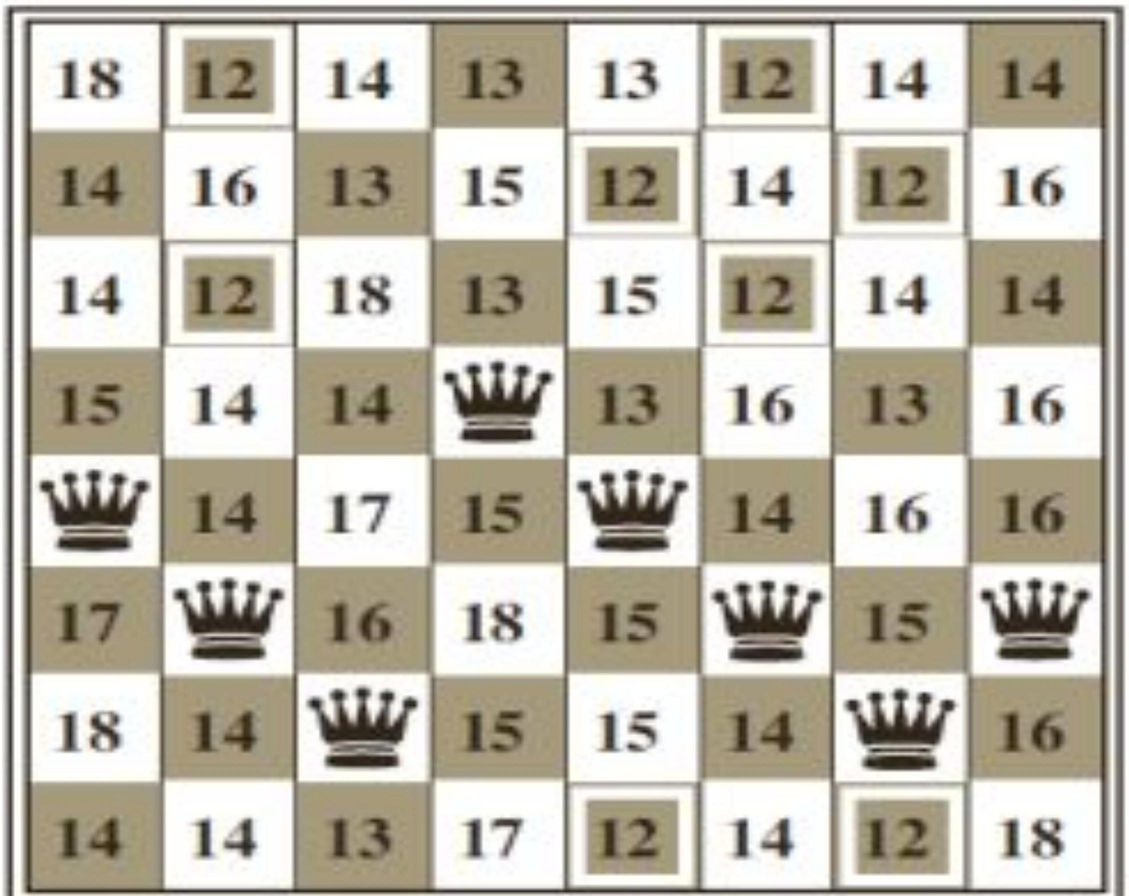
Contd..



- Almost always solves n-queens problems almost instantaneously for very large n, e.g., n = 1 million.

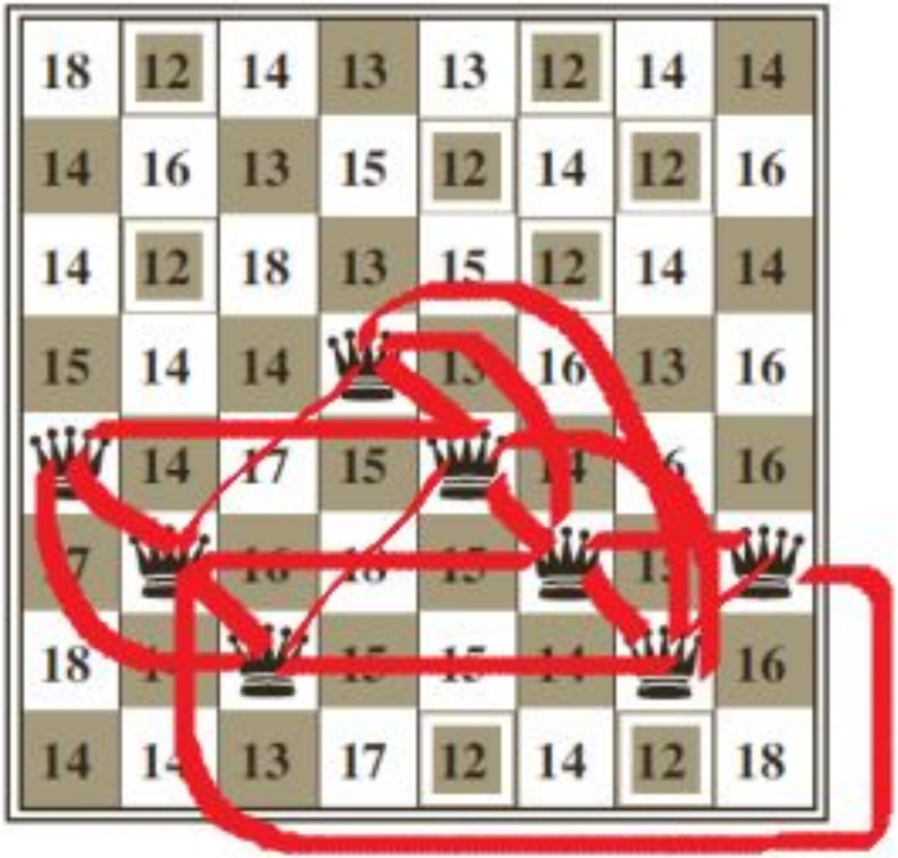
Task

- Put n queens on an $n \times n$ board with no two queens on the same row, column, or diagonal



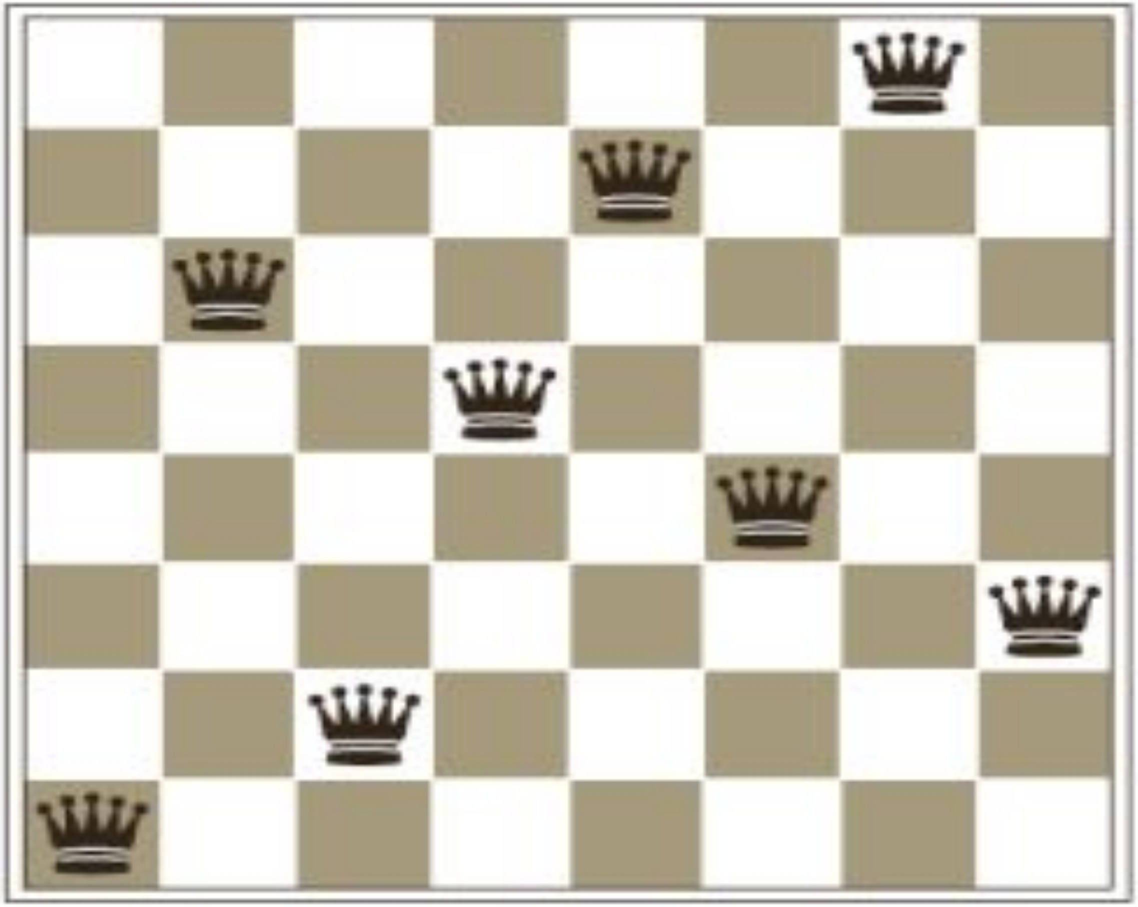
Example 2: Contd..

- An 8-queens state with heuristic cost estimate h=17
- The value of h for each possible successor obtained by moving a queen within its column.



Solution

- A local minimum in the 8-queens state space; the state has h=1.

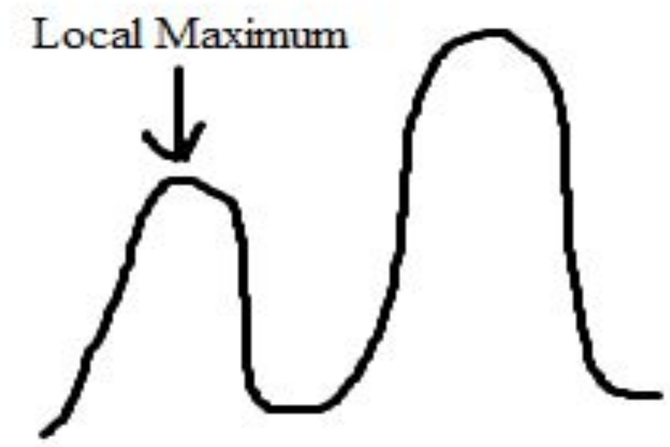


Problems in Hill Climbing Algorithm

- Although greedy algorithms often perform well, hill climbing gets stuck when:
  - Local maxima/minima
  - Plateau (shoulder or flat local maxima/minima)
  - Ridges

1. Local Maximum

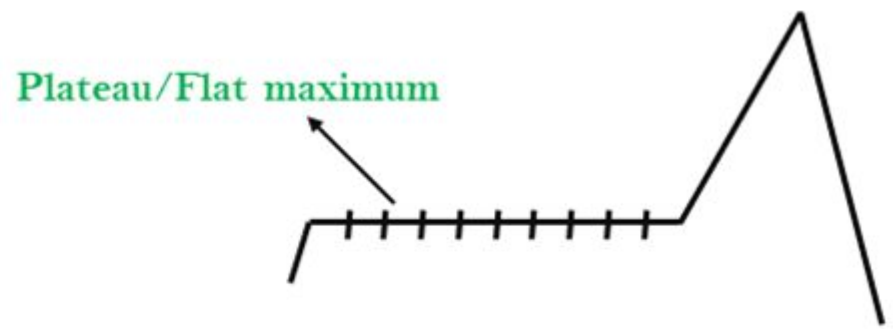
- A local maximum is a peak state in the landscape which is better than each of its neighboring states, but there is another state also present which is higher than the local maximum.



- **Solution: Backtracking technique can be a solution** of the local maximum in state space landscape. Create a list of the promising path so that the algorithm can backtrack the search space and explore other paths as well.

2. Plateau

- A plateau is the flat area of the search space in which all the neighbor states of the current state contains the same value, because of this algorithm does not find any best direction to move. A hill-climbing search might be lost in the plateau area.



- **Solution:** The solution for the plateau is to **take big steps or very little steps while searching**, to solve the problem. Randomly select a state which is far away from the current state so it is possible that the algorithm could find non-plateau region.

3. Ridges

- A special form of the local maximum. It has an area which is higher than its surrounding areas, but itself has a slope, and cannot be reached in a single move

- **Solution:** With the use of **bidirectional search**, or by moving in different directions, we can improve this problem.

Variants of Hill Climbing Algorithm

- Stochastic hill Climbing
  - It does not examine for all its neighbor before moving.
  - Selects one neighbor node at random and decides whether to choose it as a current state or examine another state
- First-Choice Hill Climbing
  - generate successors randomly until one is better than the current
  - good when a state has many successors
- Random-Restart Hill Climbing
  - conducts a series of hill climbing searches from randomly generated initial states, stops when a goal is found
  - It’s complete with probability approaching

END