AI 2002
Artificial Intelligence

Dr. Hashim Yasin

Beyond Classical Search

Beyond Classical Search

- We have addressed a single category of problems: observable, deterministic, known environments
  - where the solution is a sequence of actions

- The search algorithms that we have seen so far are designed to explore search spaces systematically.
  - When a goal is found, the ***path*** to that goal also constitutes a ***solution*** to the problem.

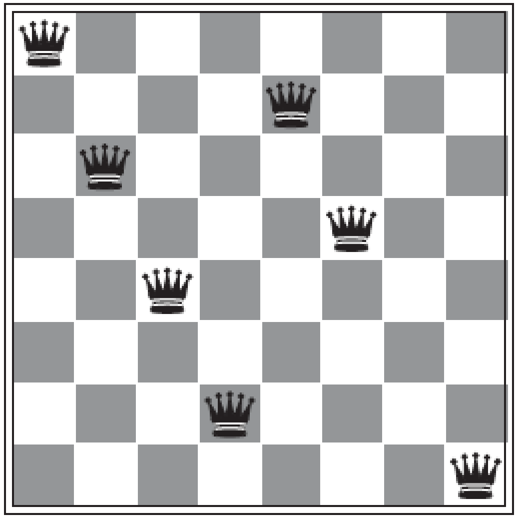
What happens when these assumptions are *relaxed?*

Beyond Classical Search

- In many problems, however, the **path to the goal is irrelevant**.

For example:

- 8-queens problem
  - *what matter, is the final configuration of queens, not the order*.
- The factory-floor layout problem
- The vehicle routing problem



Local Search Algorithms

Local Search Algorithms

- Operate using a **single current node** and generally **move only to neighbors** of that node.
  - *No concern with paths* followed by the search
  - They are *NOT systematic*

- Local search algorithms have **<u>two key advantages</u>**:
  - they use very *little memory*
  - they can often find reasonable *solutions in large* or *infinite (continuous)* state spaces

Local Search Algorithms

- The local search algorithms are useful for solving pure **optimization problems**,

Optimization problems:

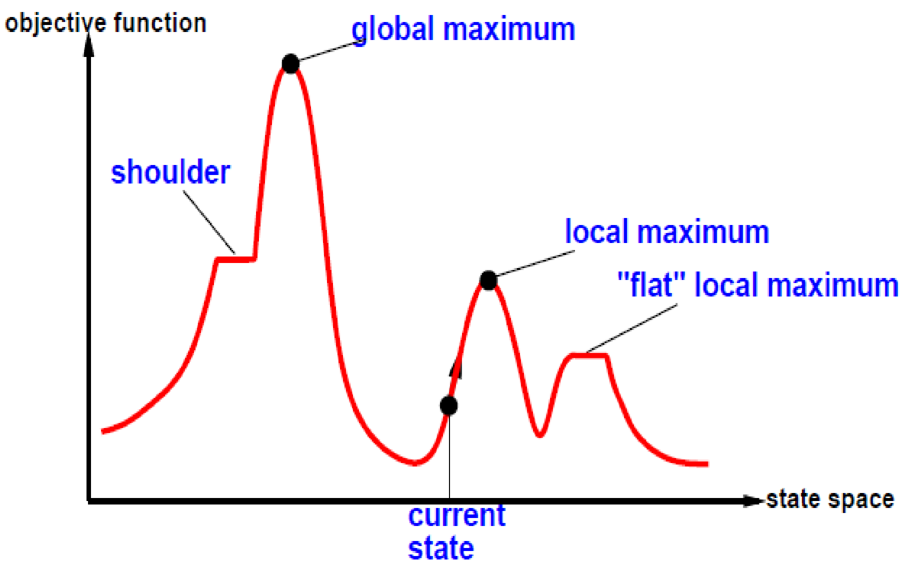
- the aim is **to find the best state** according to an *objective function*.
- The *standard search methods do not fit well* with optimization problems.

Local Search Algorithms

State-space landscape

A landscape has both

- “**location**”
  - defined by the state
- “**elevation**”
  - defined by the value of the **heuristic cost function** or **objective function**.



Local Search Algorithms

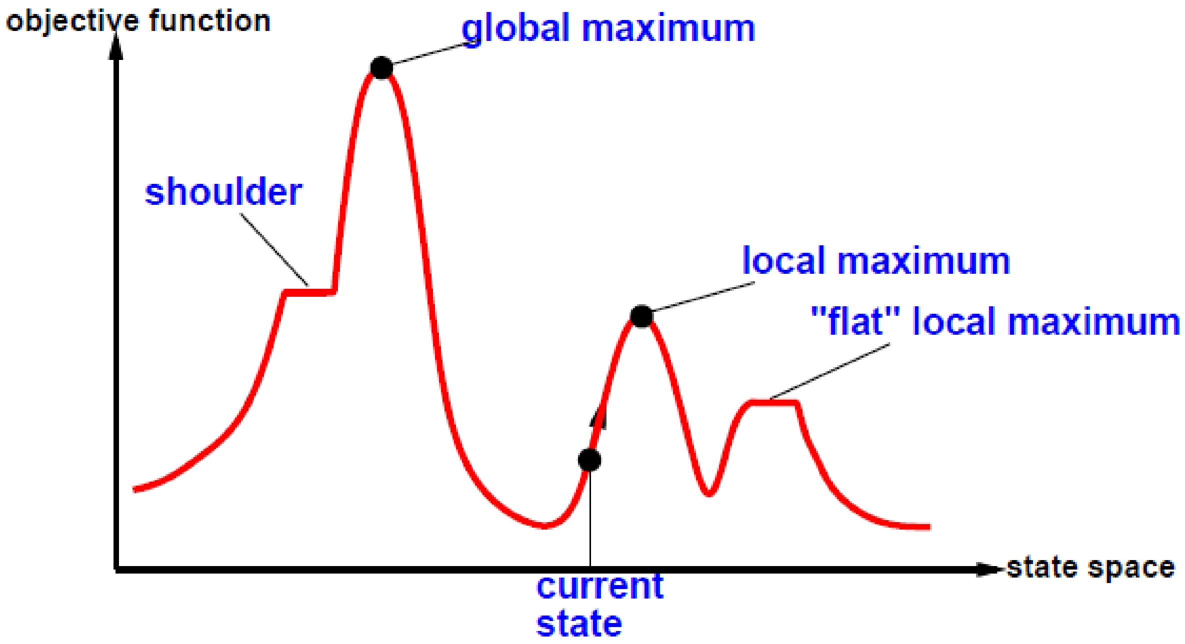
State-space landscape

Goal/Aim:

- If elevation corresponds to **cost**, then the aim is to find the lowest valley—a **global minimum**;
- If elevation corresponds to an **objective function**, then the aim is to find the highest peak—a **global maximum**.

Local Search Algorithms

State-space landscape



A one-dimensional state-space landscape in which *elevation corresponds to the objective function*. The aim is to find the global maximum.

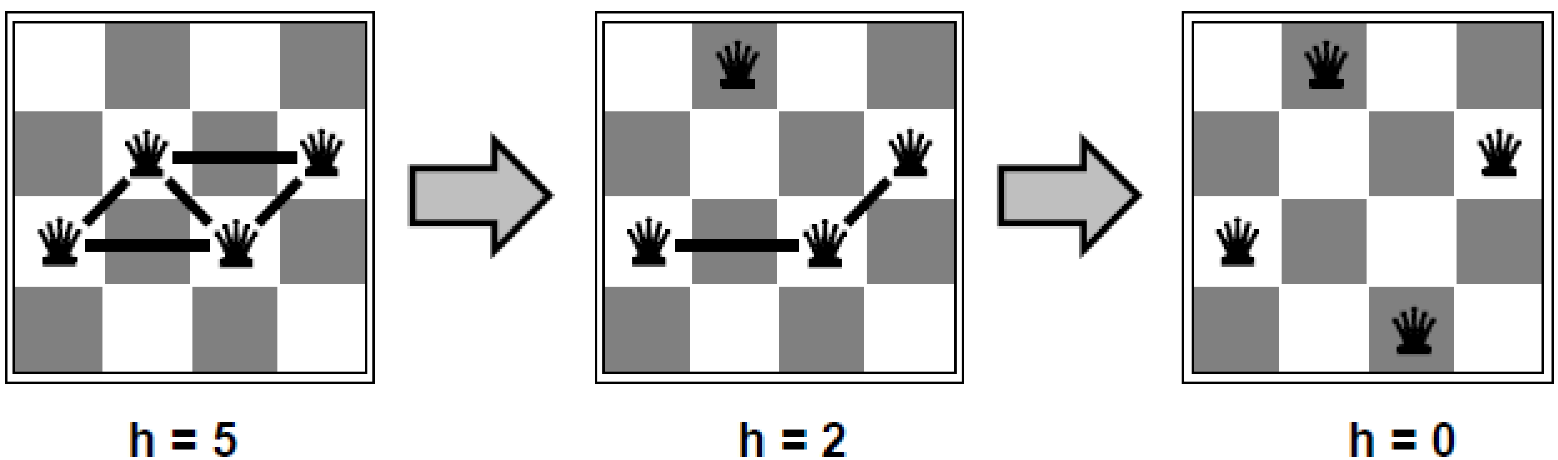
Local Search Algorithms

- A **complete local search** algorithm always finds a goal if one exists;
- An **optimal** algorithm always finds a global minimum/maximum.

- Local search algorithms typically use a **complete-state formulation**,
  - In 8-queen problem, where *each state has 8 queens* on the board, one per column.

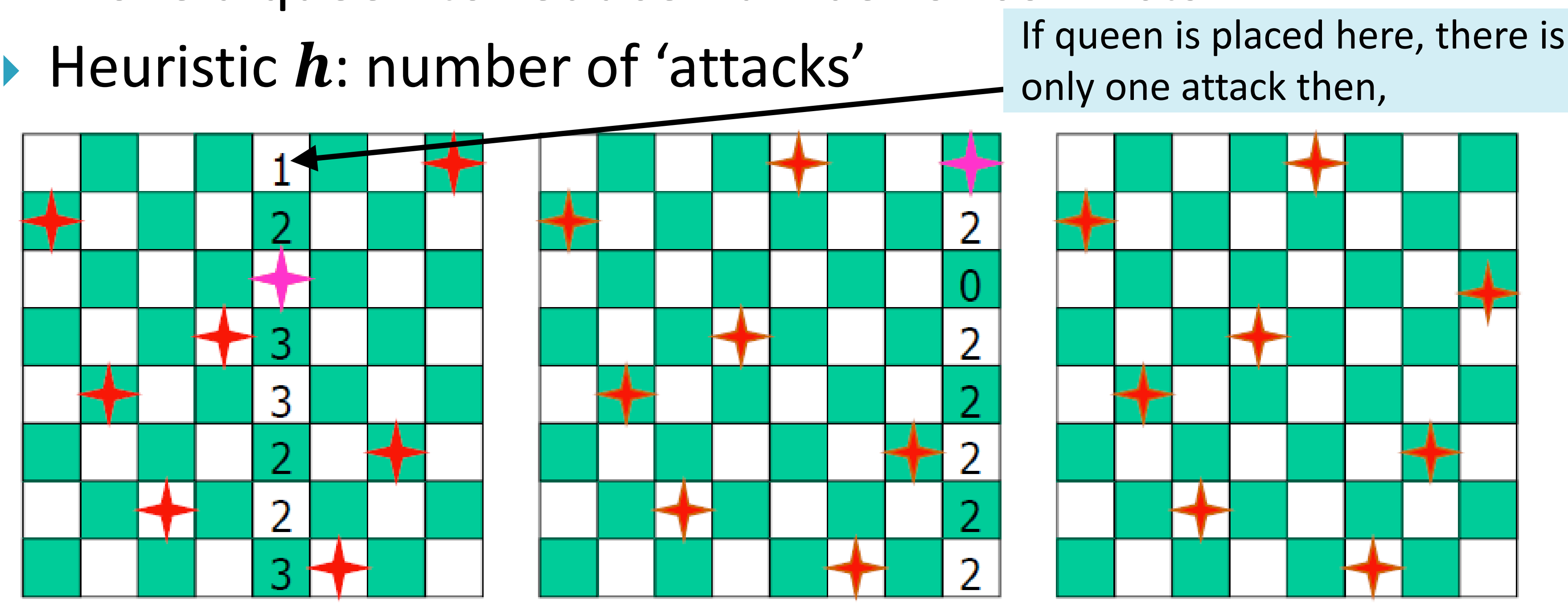
Example: n-queens Problems

- Put $\boldsymbol{n}$ queens on an $\boldsymbol{n} \times \boldsymbol{n}$ board with no two queens on the same row, column, or diagonal.
- Move a queen to reduce number of conflicts
- Heuristic $\boldsymbol{h}$: number of 'attacks'



Example: n-queens Problems

- Put $\boldsymbol{n}$ queens on an $\boldsymbol{n} \times \boldsymbol{n}$ board with no two queens on the same row, column, or diagonal
- Move a queen to reduce number of conflicts
- Heuristic $\boldsymbol{h}$: number of 'attacks'



Hill-climbing Search

Hill-climbing Search

- It is simply a **loop** that continuously moves in the direction of increasing value—that is, uphill.
- It terminates when it reaches a "**peak**" where no neighbor has a higher value.
  - *does not maintain a **search tree***
  - need only
    - *record the state* and
    - the value of the *objective function*.
- Hill climbing only looks towards the immediate neighbors of the current state.

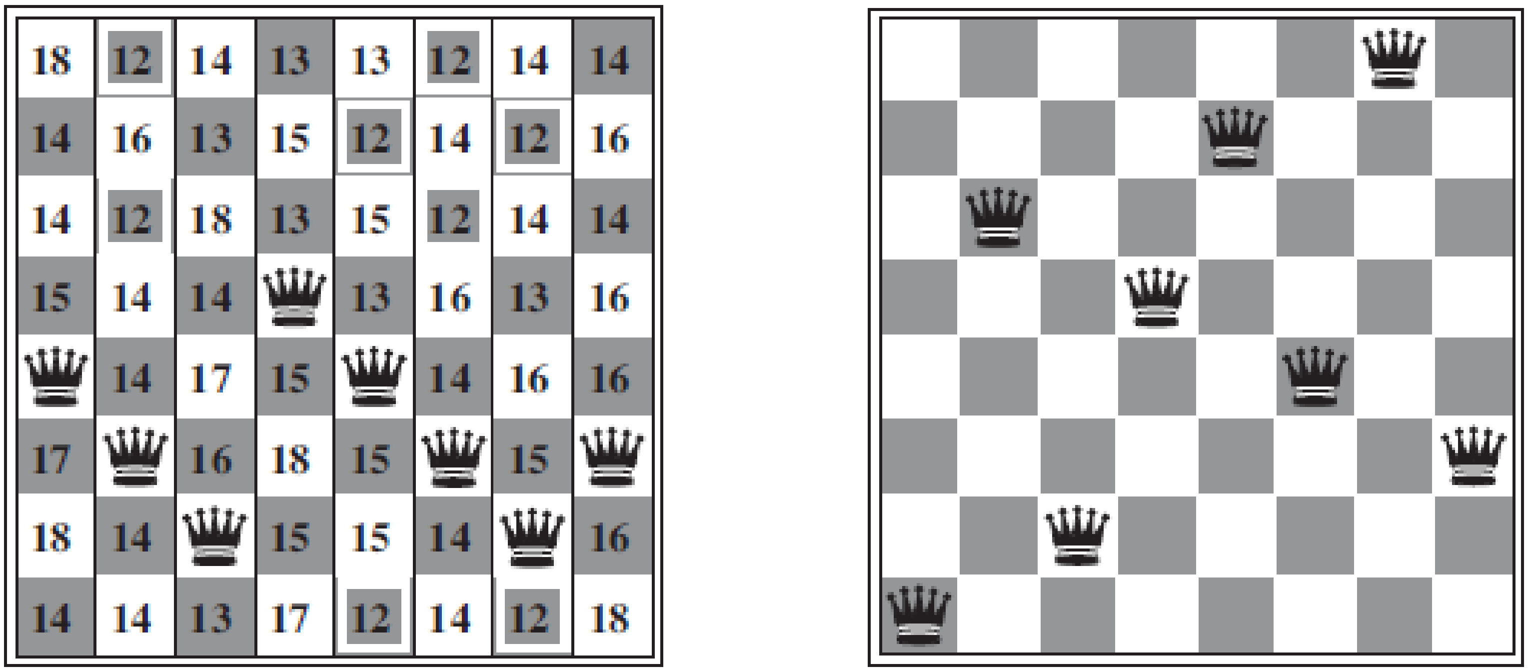
Hill-climbing Search

function HILL-CLIMBING(*problem*) **returns** a state that is a local maximum

  current ← MAKE-NODE(*problem*.INITIAL-STATE)
  loop do
    neighbor ← a highest-valued successor of *current*
    if neighbor.VALUE ≤ current.VALUE **then return** *current*.STATE
    current ← *neighbor*

- At *each step the current node is replaced by the best neighbor*; the neighbor with the highest VALUE,
- If a **heuristic cost estimate *h*** is used, we would find the neighbor with the ***lowest h***.

Hill-climbing Search



(a) An 8-queens state with heuristic cost estimate $h = 17$, showing the value of h for each possible successor. (b) A local minimum: the state has $h = 1$ but every successor has a higher cost.

Hill-climbing Search

- The hill climbing **often gets stuck** for the following reasons:

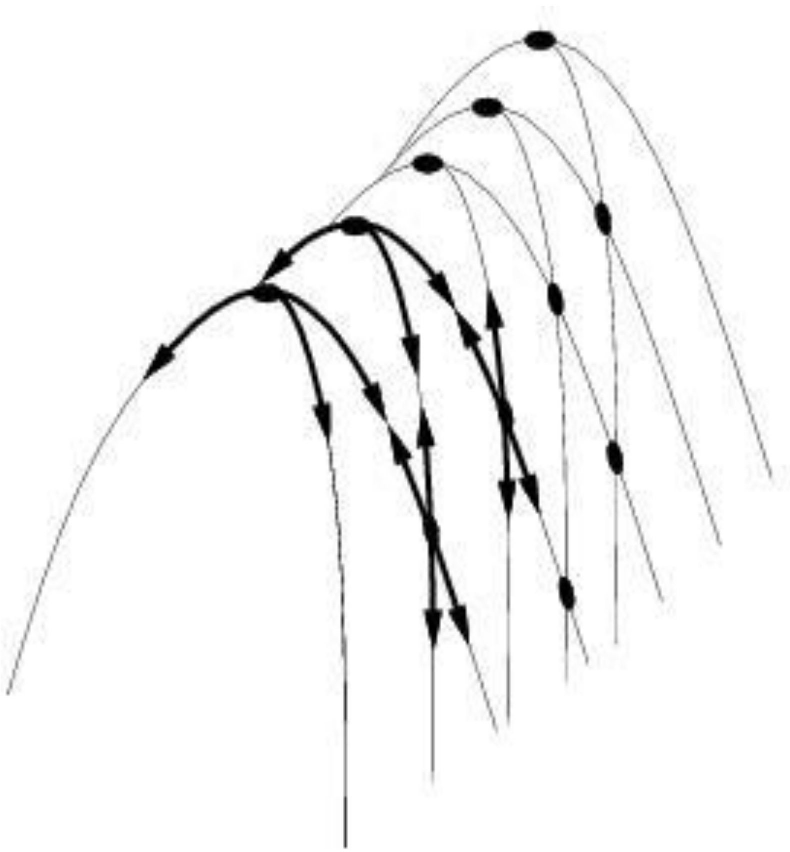
Local maxima:

- A local maximum is a peak that is higher than each of its neighboring states but lower than the global maximum.

Hill-climbing Search

Ridges:

- A sequence of local maxima
- *Its difficult for greedy algorithms to navigate*



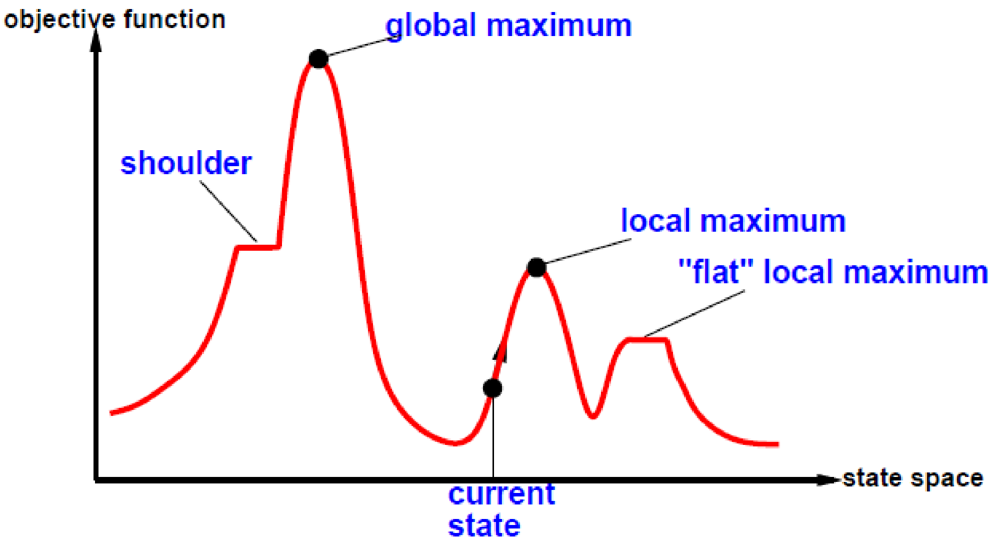
Hill-climbing Search

Plateaux:

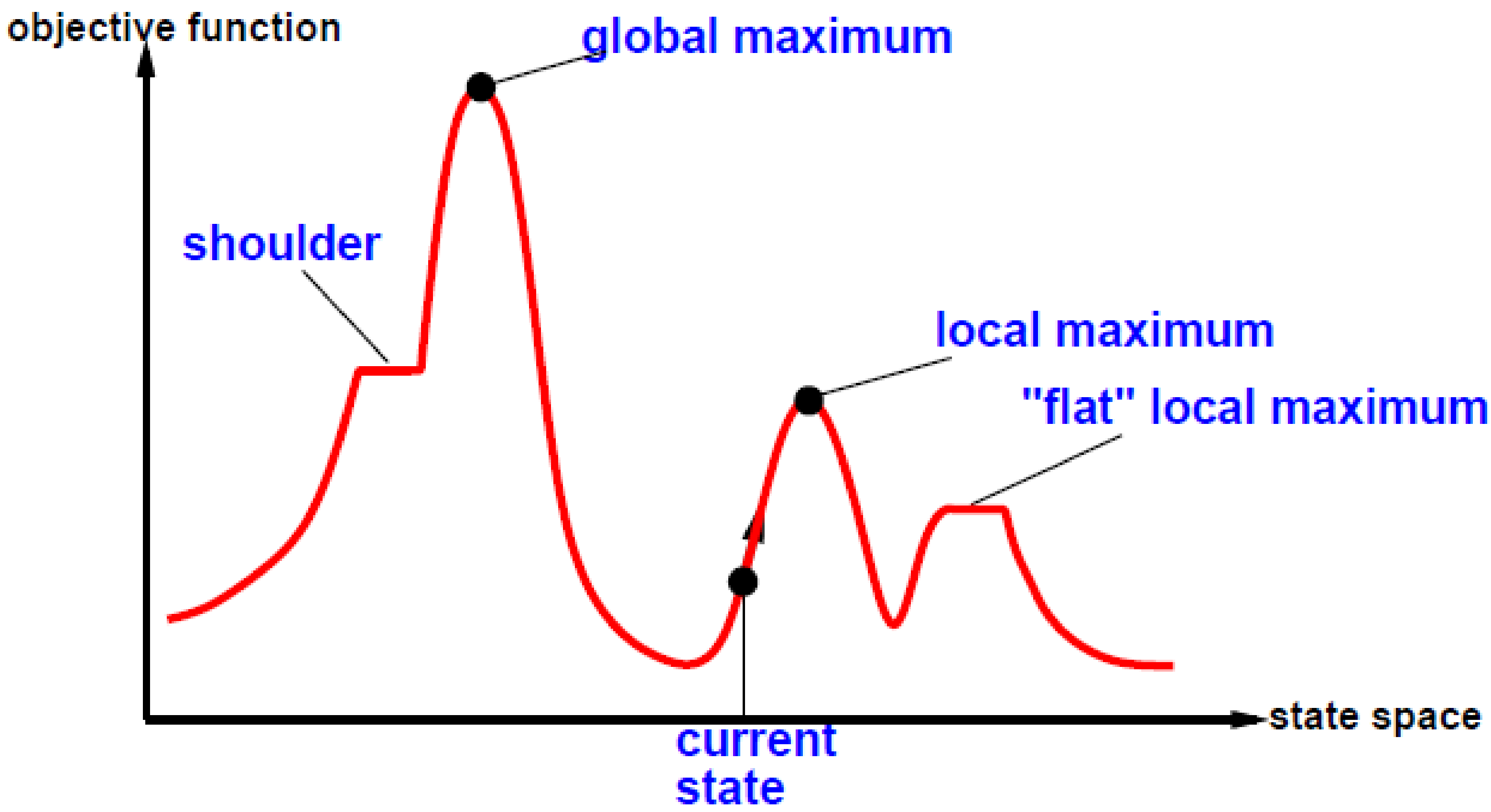
- An area of the state space where the **evaluation function is flat**.
- It can be a **flat local maximum**, from which no uphill exit exists,

Shoulder:

- from which progress is possible



Hill-climbing Search



Reading Material

- **Artificial Intelligence,** A Modern Approach

  Stuart J. Russell and Peter Norvig

  - **Chapter 3 & 4.**