# Hill Climbing

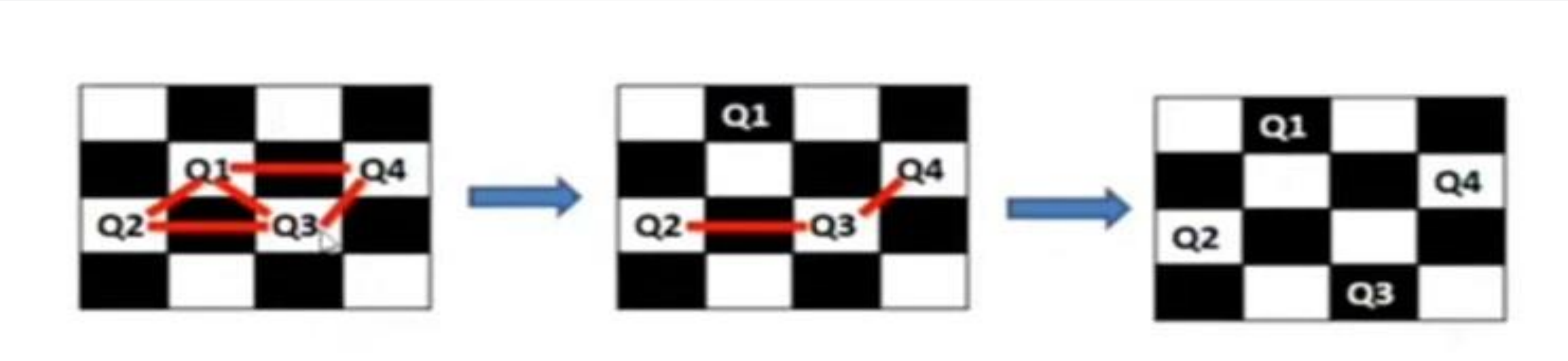
# Local Search

- *Hill climbing is a local search algorithm used for mathematical optimization. It starts with an arbitrary solution to a problem and iteratively makes small changes to the solution, choosing the modified solution that improves the objective function the most.*
- *It is uniformed and informed search algorithms that we have seen are designed to explore search spaces systematically*
- *They keep one or more paths in memory and record which alternatives have been explored at each point along the path when goal is found the path to that goal also constitutes a solution to the problem*
- *In many problem like n queen problem the path to the goal is irrelevant. If the path to goal does not matter, then consider a different class of algorithms that do not worry about paths at all.*

- *Local search algorithms operate using a single current node and generally move only to neighbors of that node.*
- *No search tree or graph*
- *They use very little memory(usually a constant amount)*
- *They can often find reasonable solutions in large or infinite state spaces and 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 paths to goal is irrelevant and the goal state itself is the solution.*

# N-Queen problem

- Put n-queens on an n*n board with no two queens on the same row, same column , or diagonal
- Move a queen to reduce number of conflicts.
- Objective function: number of conflicts
- Eg. 4-Queen problem



# Hill Climbing Algorithm terminologies

State-space Landscape of Hill climbing algorithm:

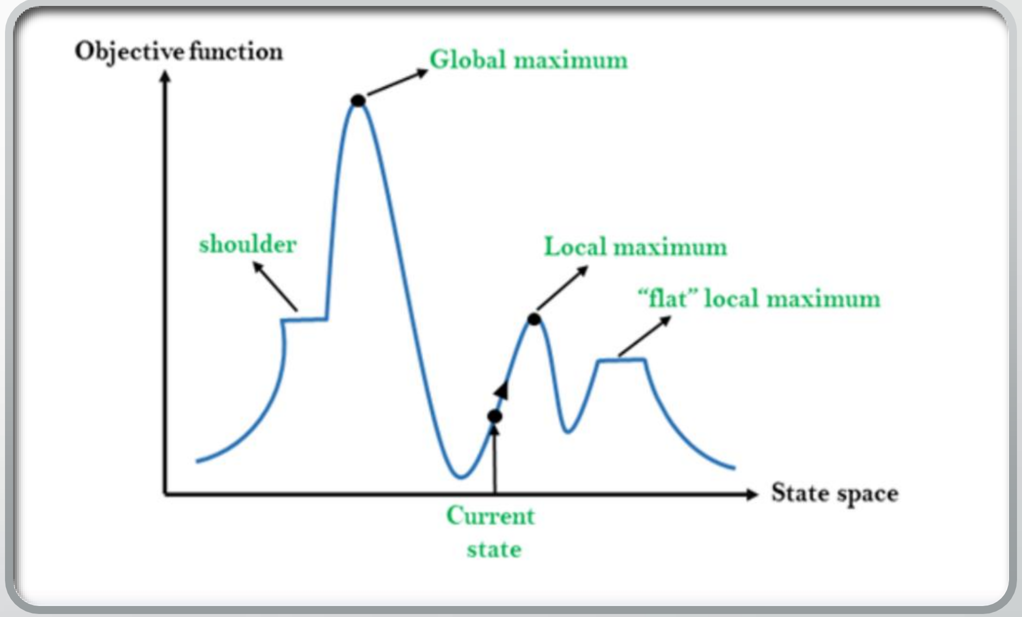
**Gobal Maximum:** It is the highest point on the hill ,which is the goal state

**Local Maximum:** It is the peak higher than all other peaks but lower than the global maximum.

**Flat local maximum**: It is flat area over the hill where it has no uphill or downhill. It is a saturated point of the hill.

**Shoulder:** It is also a flat area where the summit is possible.

**Current state**: It is the current position of the person.



# Hill-Climbing generate-And-test + heuristics

**Generate-And-Test Algorithm**

1.Define current state as an initial state

2. Apply any possible operation on the current state and generate a possible

Solution

3. Compare newly generated solution with the goal state

4. If the goal is achieved or no new states can be created, quit. Otherwise, return to the step 2

-It works very well with simple problems

-As it is an exhaustive search, it is not feasible to consider it while dealing with large problem spaces.

# Hill-Climbing algorithm

- Define the current state as an initial state
- Loop until the goal state is achieved or no more operators can be applied on the current state:

  - Apply an operation to current state and get a new state

  -Compare the new state with the goal

  -Quit if the goal state is achieved

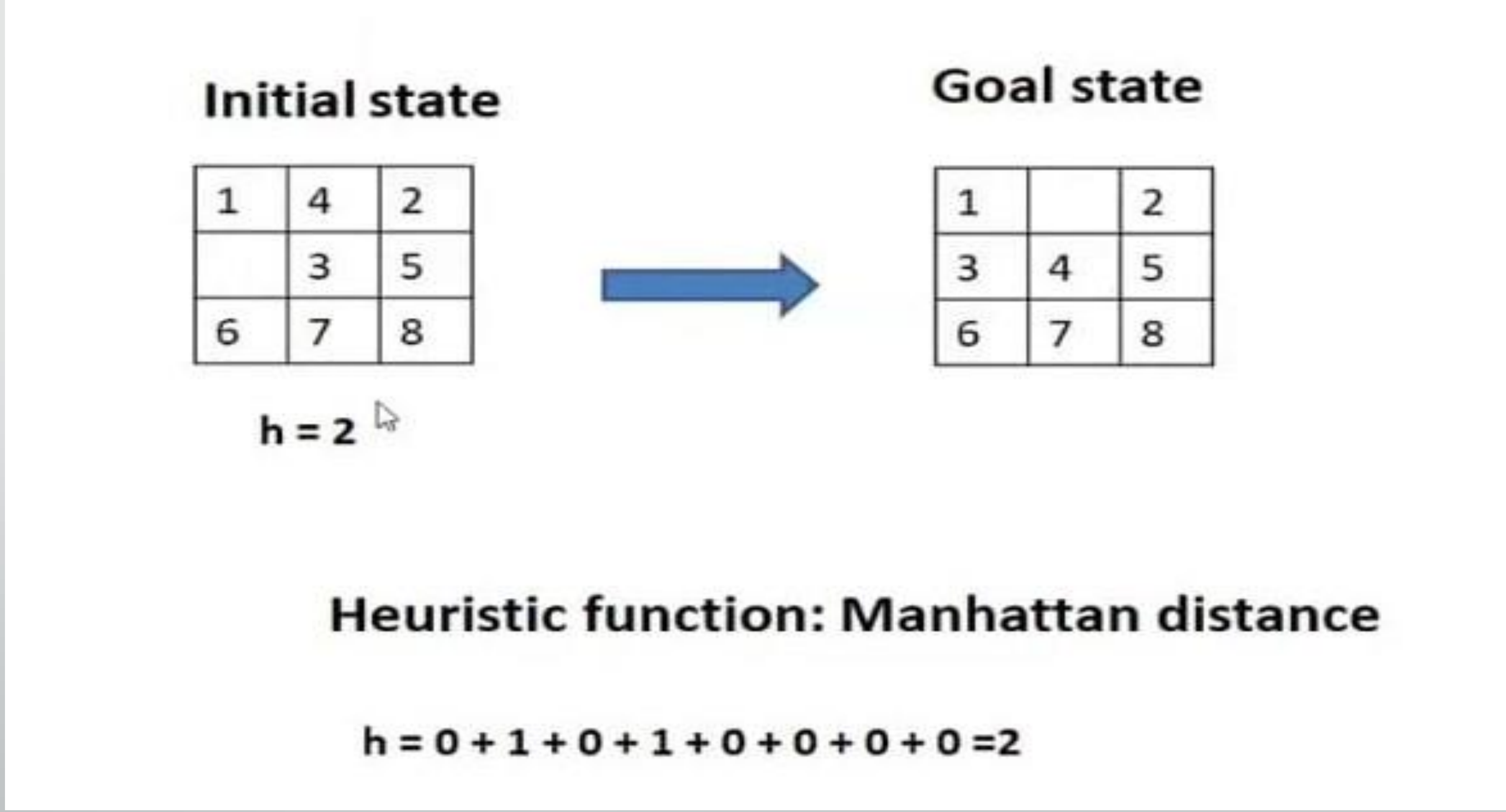
  -Evaluate new state with heuristic function and compare it with the current state

  -If the newer state is closer to the goal compared to current state, update the current state
- It reaches the goal state with iterative improvements.
- In Hill-Climbing algorithm, finding goal is equivalent to reaching the top of the hill.

# Hill-Climbing algorithm

- The algorithm does not maintain a search tree, so the data structure for the current node only need to record the state and the value of the objective function.
- It does not look ahead beyond the immediate neighbors of the current state
- It is sometimes called greedy local search because it grabs a good neighbor state without thinking ahead about where to go next.
- Greedy algorithms often perform quite well and Hill climbing often makes rapid progress toward a solution.

# 8-puzzle problem using hill climbing



Initial state

|   |   |   |
|---|---|---|
| 1 | 4 | 2 |
|   | 3 | 5 |
| 6 | 7 | 8 |

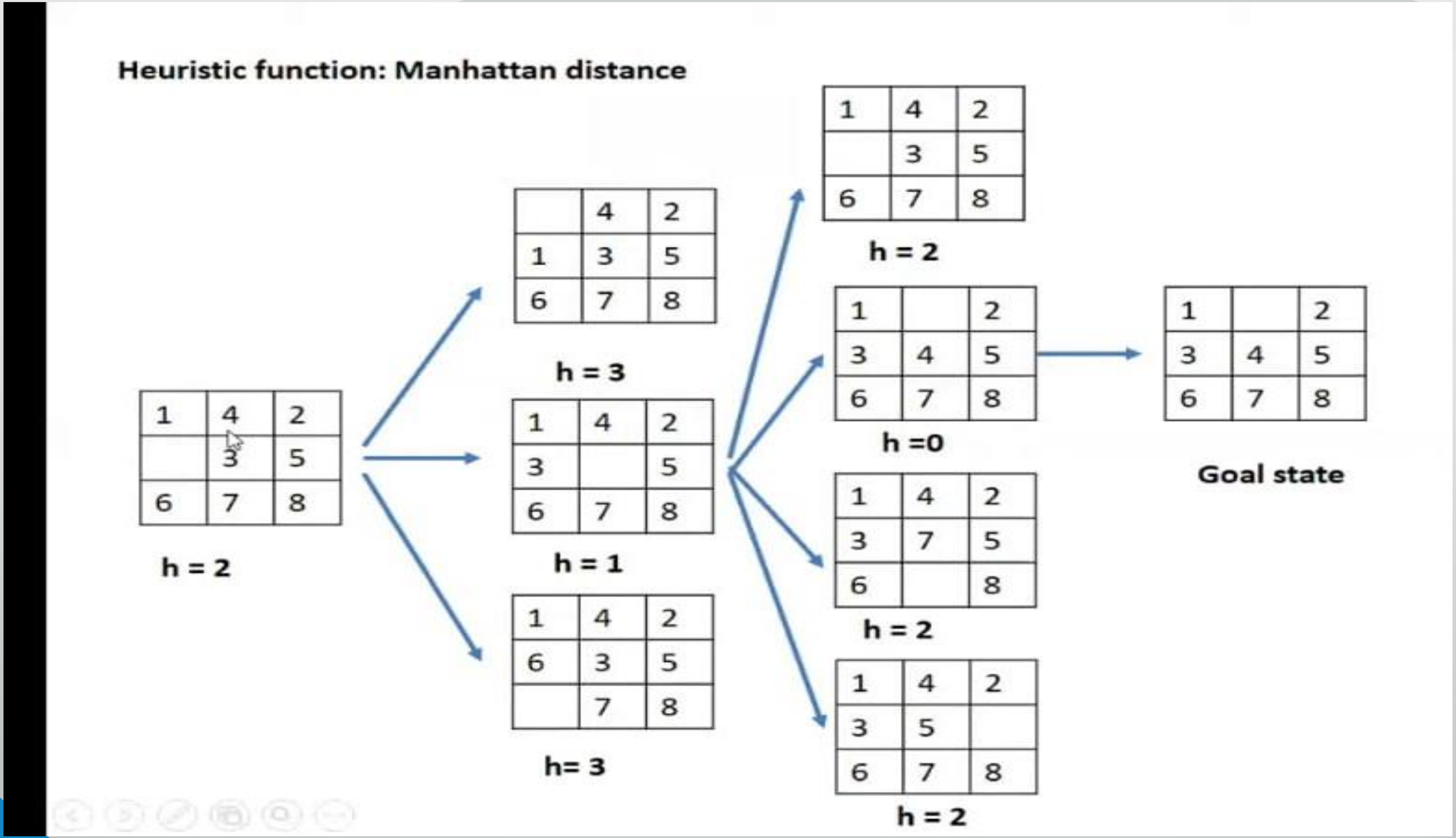
h = 2

Goal state

|   |   |   |
|---|---|---|
| 1 |   | 2 |
| 3 | 4 | 5 |
| 6 | 7 | 8 |

Heuristic function: Manhattan distance

h = 0 + 1 + 0 + 1 + 0 + 0 + 0 + 0 =2



Heuristic function: Manhattan distance

|   |   |   |
|---|---|---|
| 1 | 4 | 2 |
|  | 3 | 5 |
| 6 | 7 | 8 |

h = 2

|   |   |   |
|---|---|---|
|  | 4 | 2 |
| 1 | 3 | 5 |
| 6 | 7 | 8 |

h = 3

|   |   |   |
|---|---|---|
| 1 | 4 | 2 |
| 3 |  | 5 |
| 6 | 7 | 8 |

h = 1

|   |   |   |
|---|---|---|
| 1 | 4 | 2 |
| 6 | 3 | 5 |
|  | 7 | 8 |

h= 3

|   |   |   |
|---|---|---|
| 1 | 4 | 2 |
|  | 3 | 5 |
| 6 | 7 | 8 |

h = 2

|   |   |   |
|---|---|---|
| 1 |  | 2 |
| 3 | 4 | 5 |
| 6 | 7 | 8 |

h =0

|   |   |   |
|---|---|---|
| 1 | 4 | 2 |
| 3 | 7 | 5 |
| 6 |  | 8 |

h = 2

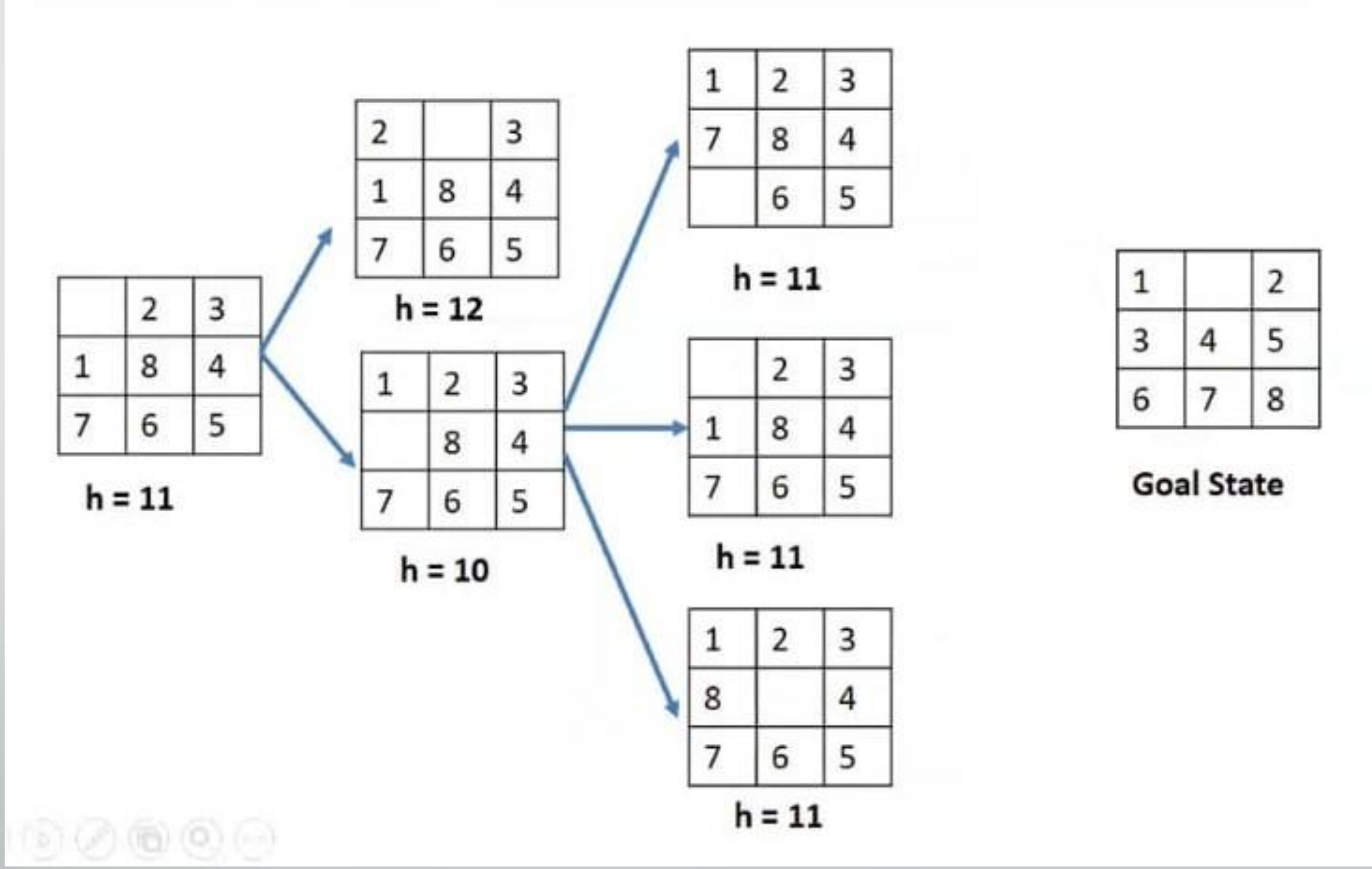
|   |   |   |
|---|---|---|
| 1 | 4 | 2 |
| 3 | 5 |  |
| 6 | 7 | 8 |

h = 2

|   |   |   |
|---|---|---|
| 1 |  | 2 |
| 3 | 4 | 5 |
| 6 | 7 | 8 |

Goal state

# Hill climbing example stuck at local maximum



|  |  |  |
|---|---|---|
|  | 2 | 3 |
| 1 | 8 | 4 |
| 7 | 6 | 5 |

h = 11

|  |  |  |
|---|---|---|
| 2 |  | 3 |
| 1 | 8 | 4 |
| 7 | 6 | 5 |

h = 12

|  |  |  |
|---|---|---|
| 1 | 2 | 3 |
|  | 8 | 4 |
| 7 | 6 | 5 |

h = 10

|  |  |  |
|---|---|---|
| 1 | 2 | 3 |
| 7 | 8 | 4 |
|  | 6 | 5 |

h = 11

|  |  |  |
|---|---|---|
|  | 2 | 3 |
| 1 | 8 | 4 |
| 7 | 6 | 5 |

h = 11

|  |  |  |
|---|---|---|
| 1 | 2 | 3 |
| 8 |  | 4 |
| 7 | 6 | 5 |

h = 11

|  |  |  |
|---|---|---|
| 1 |  | 2 |
| 3 | 4 | 5 |
| 6 | 7 | 8 |

Goal State

# Disadvantages of Hill Climbing

- Hill Climbing is a short sighted technique as it evaluates only immediate possibilities. So it may end up in few situations from which it can not pick any further states
- Hill climbing algorithm is a fast and furious approach. It finds the solution state rapidly because it is quite easy to improve a bad state.

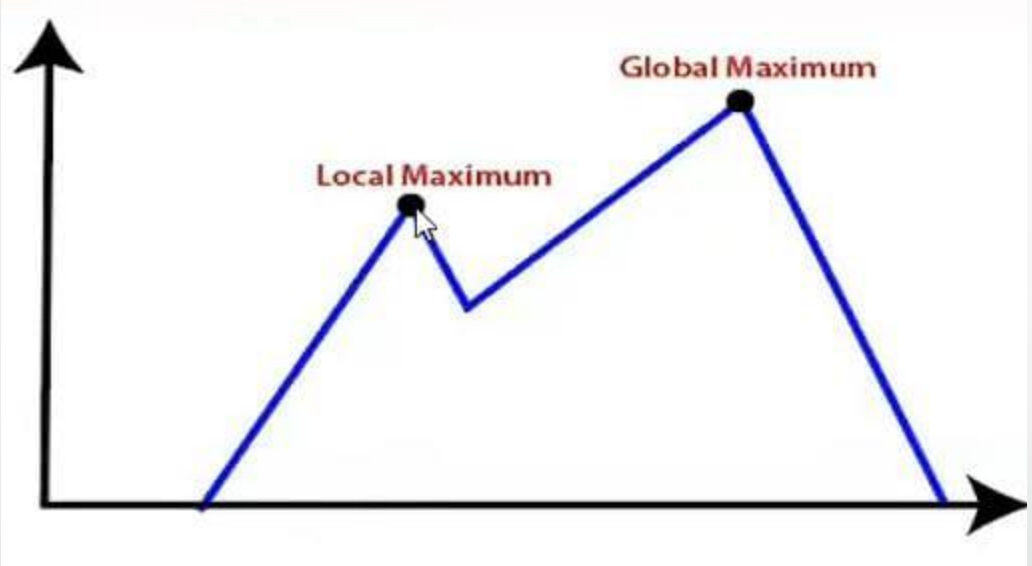
But, there are following limitations of this search:

-Local maximum

-Plateau

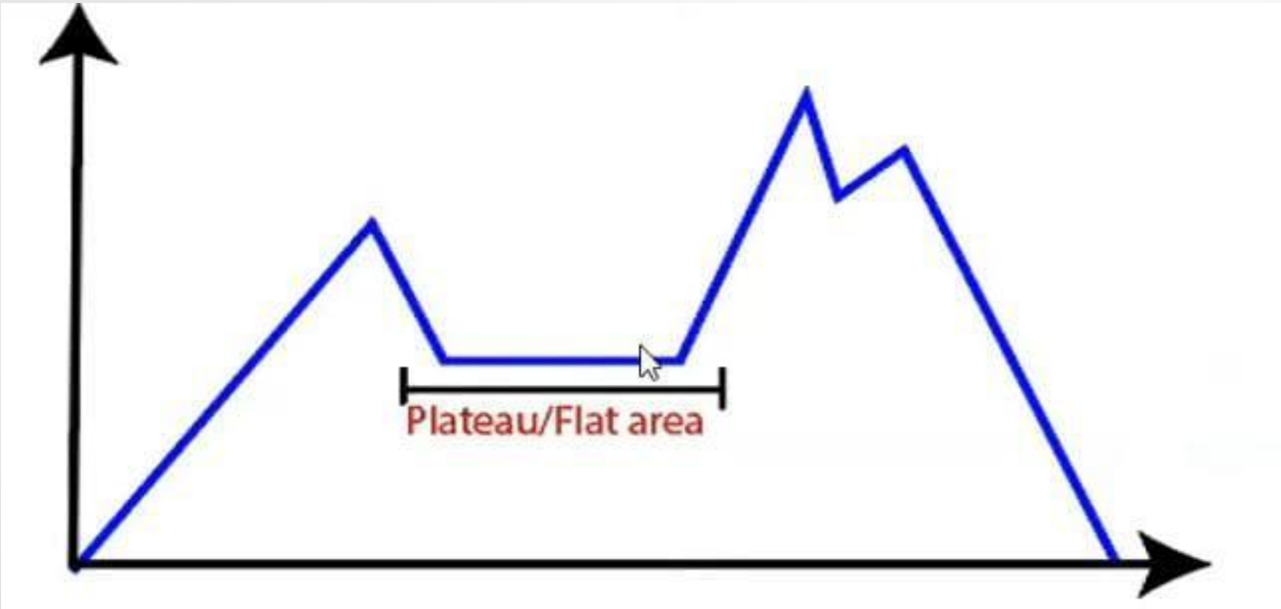
-Ridge

# Local maximum



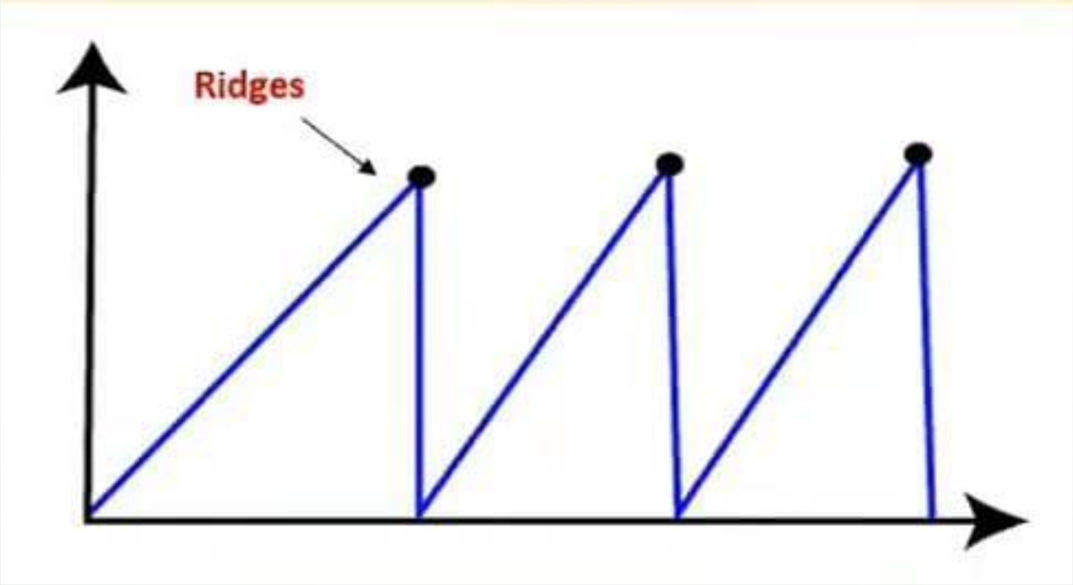
- It’s a state which is better than all neighbors, but there exists a better state which is far from the current state; if local maximum occurs within sight of the solution , it is known as "foothills".

# Plateau



- In this state, all neighboring states have same heuristic values, so its unclear to choose the next state by making local comparisons

# Ridges



- It is a challenging problem where the person finds two or more local maxima of the same height commonly. It becomes difficult for person to navigate the right point and stuck to that point itself.

# Limitations

- Even though hill climbing technique is much better than exhaustive search, it's still not optimal in large problem spaces.
- We can always encode global information into heuristic functions to make smarter decisions, but then computational complexity will be much higher than it was earlier.
- Hill climbing algorithm can be very beneficial when clubbed with other techniques.