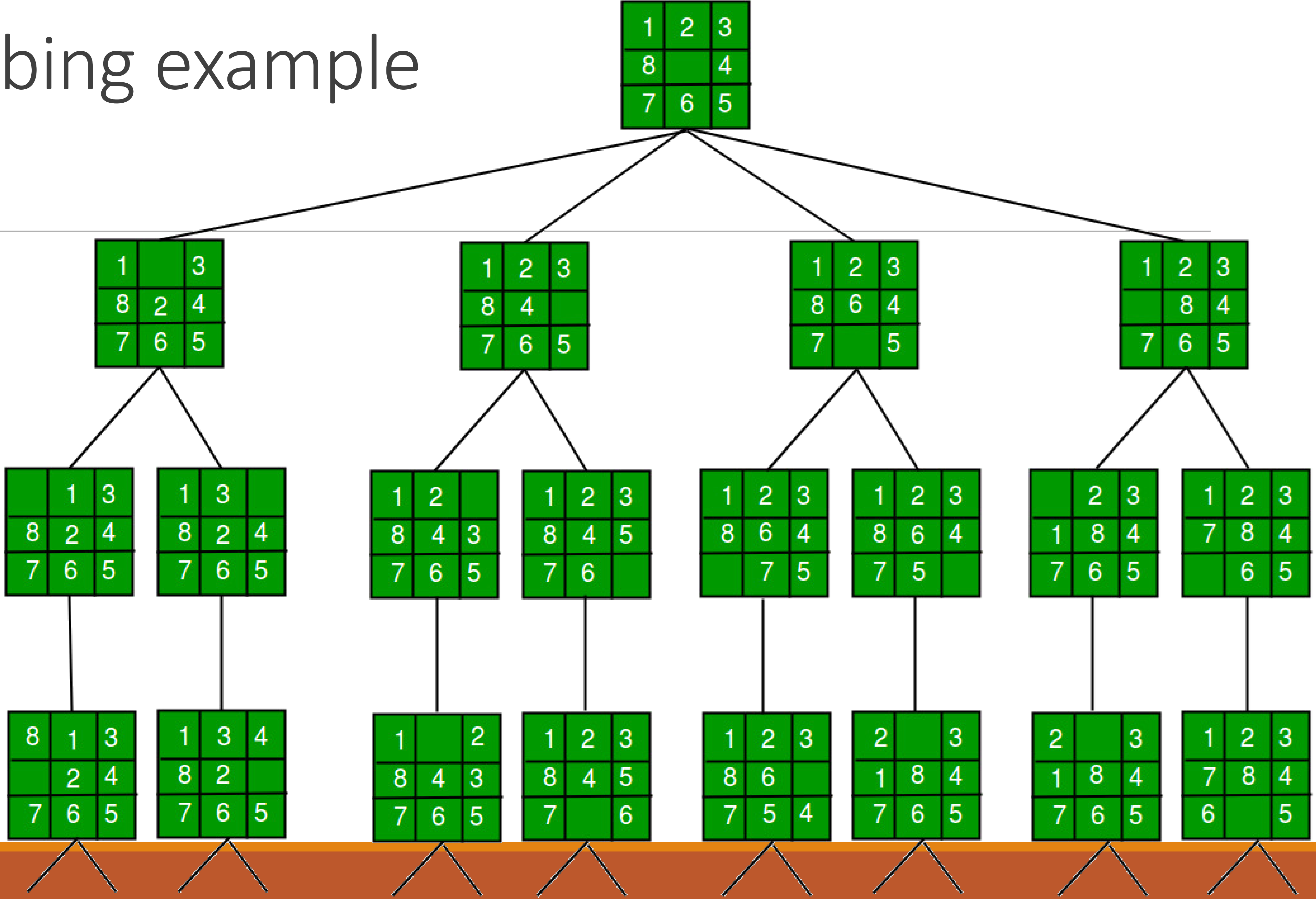
SEARCH IN COMPLEX ENVIRONMENTS

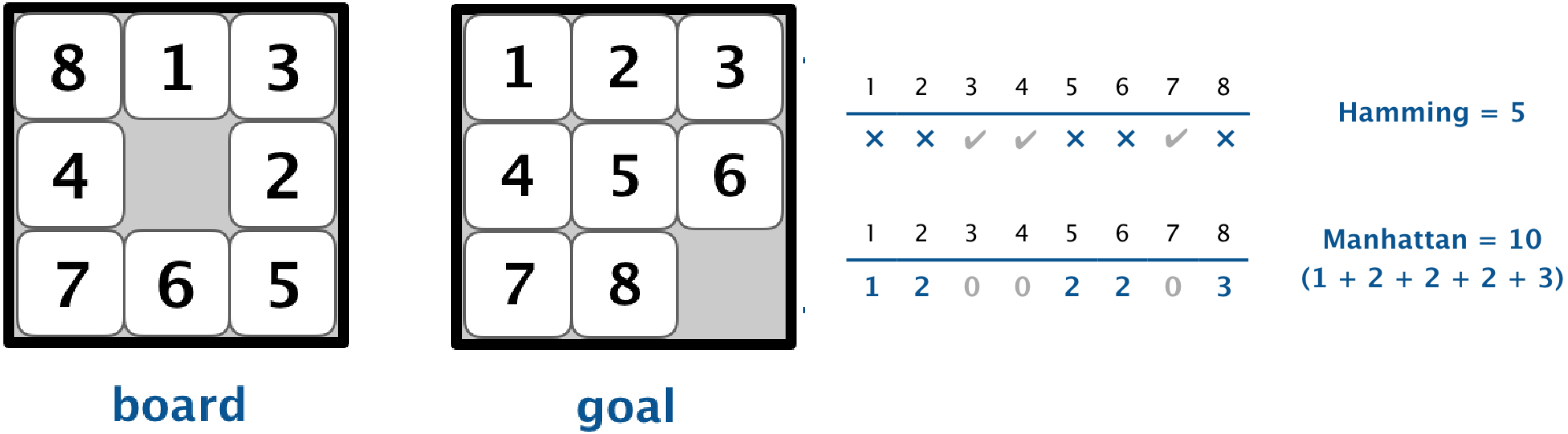
الدكتور/ مصطفى شبل

Hill climbing example

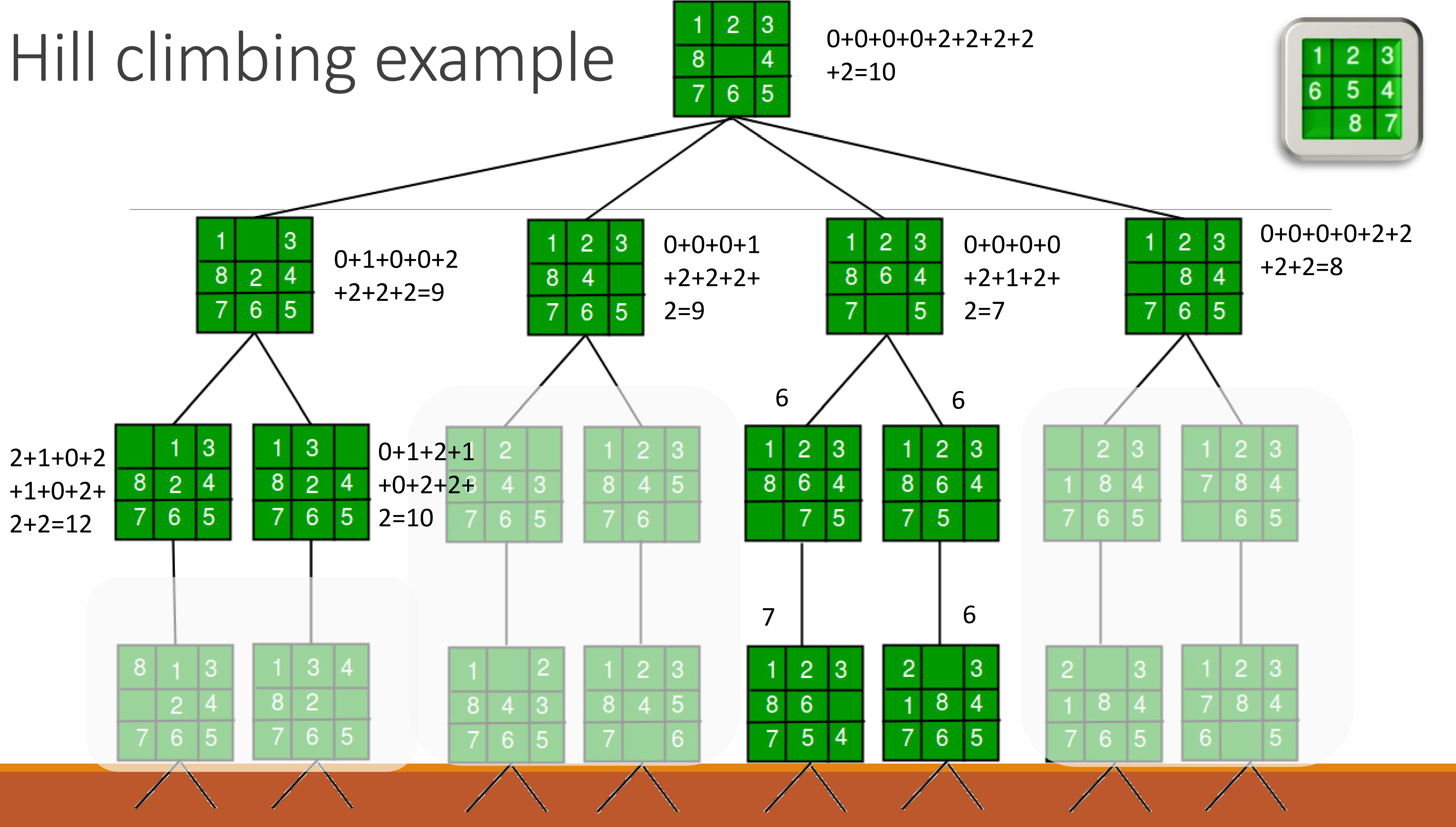


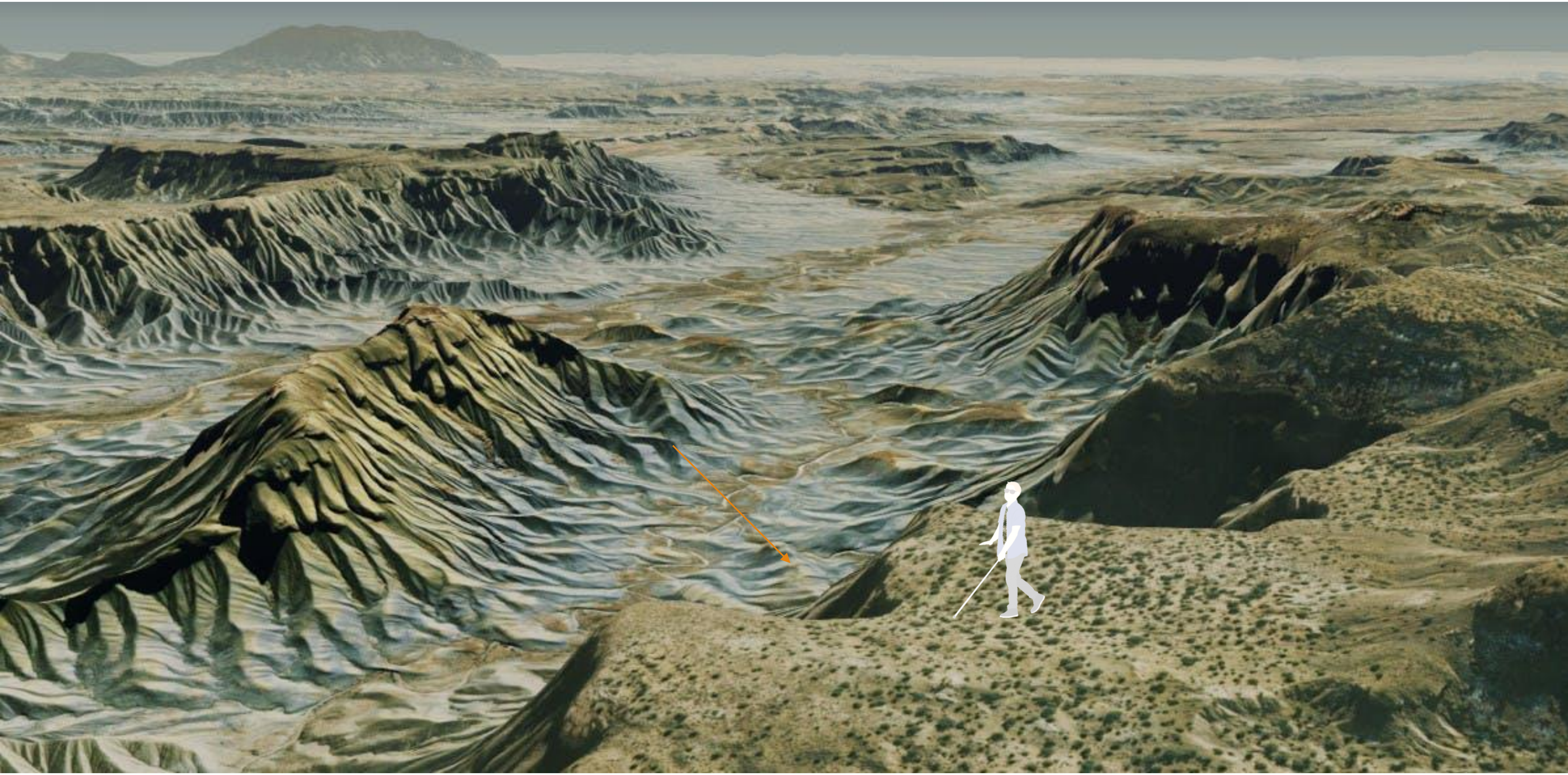
Manhattan Distance

Manhattan distance is a distance metric between two points in a N dimensional vector space



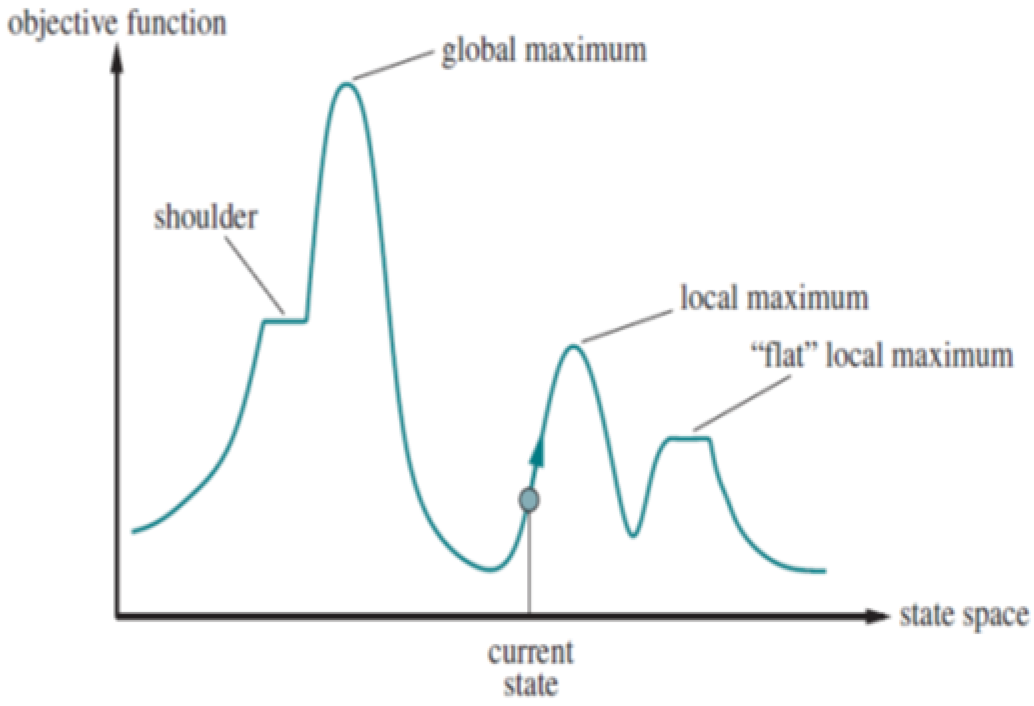
Hill climbing example

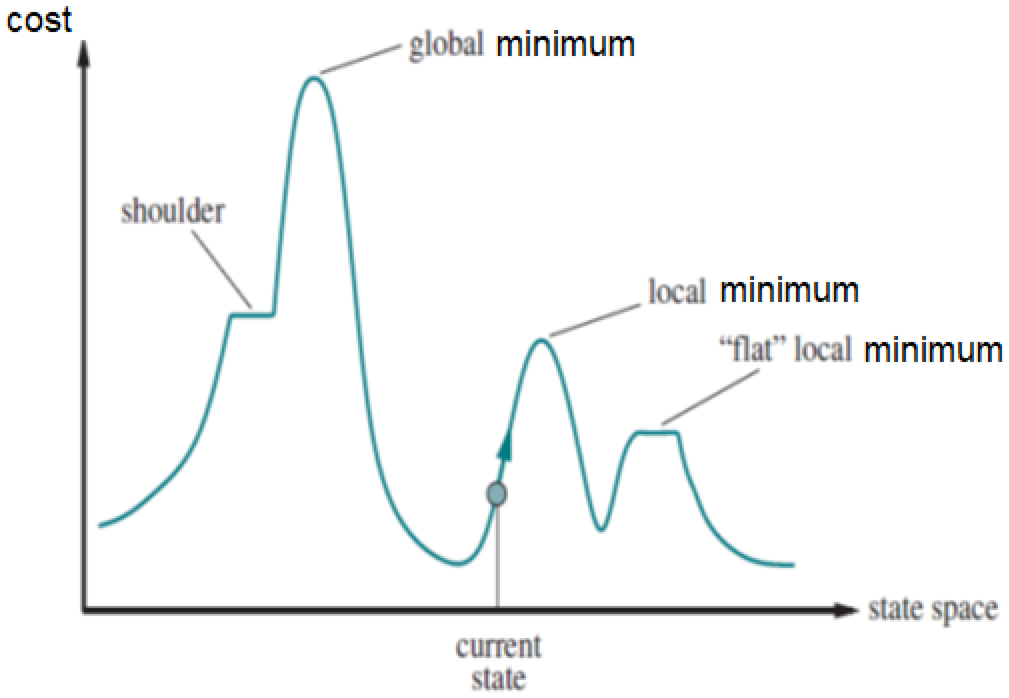




Local Search and Optimization Problems

<u>Local search algorithms:</u> operate by searching from a start state to neighboring states, without keeping track of the paths, nor the set of states that have been reached.



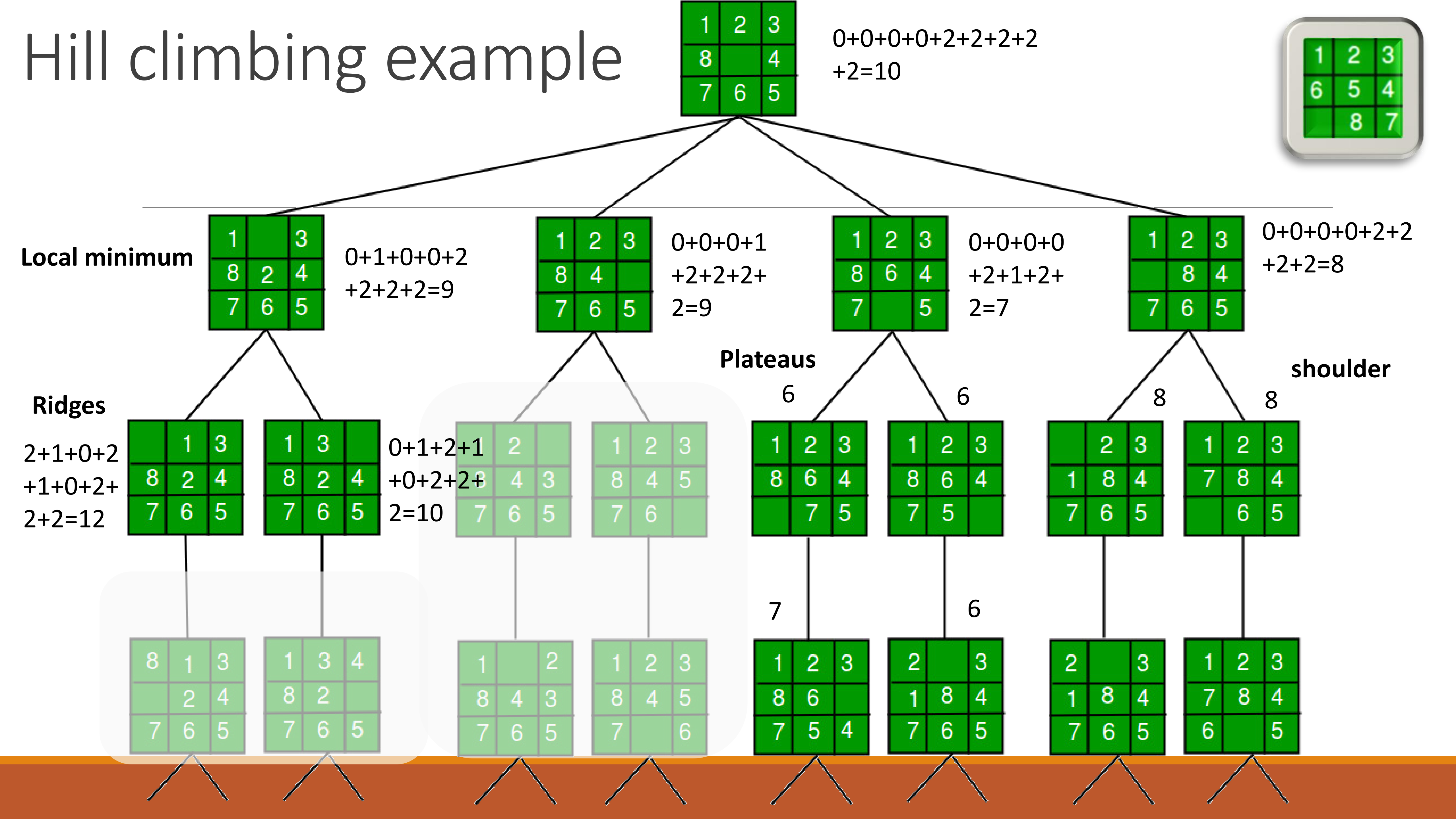


Definitions

Ridges: a sequence of local maxima that is very difficult for greedy algorithms to navigate

Plateaus: A plateau is a flat area of the state-space landscape

Hill climbing example



Pseudo code (greedy local search)

function HILL-CLIMBING(*problem*) **returns** a state that is a local maximum
    current ← *problem*.INITIAL
    while *true* **do**
        neighbor ← a highest-valued successor state of *current*
        if VALUE(*neighbor*) ≤ VALUE(*current*) **then return** *current*
        current ← *neighbor*

Local search and optimization

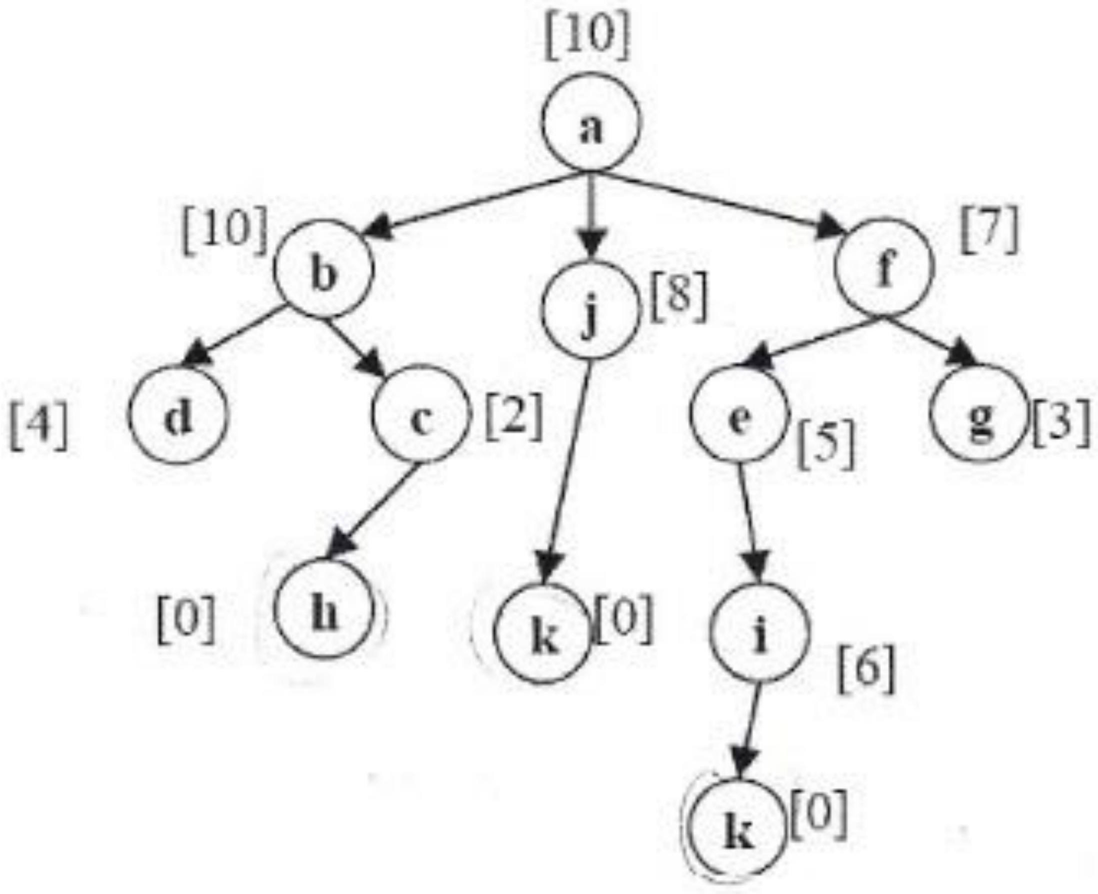
Local search

- Keep track of single current state
- Move only to neighboring states
- Ignore paths
- Use little space

Advantages:

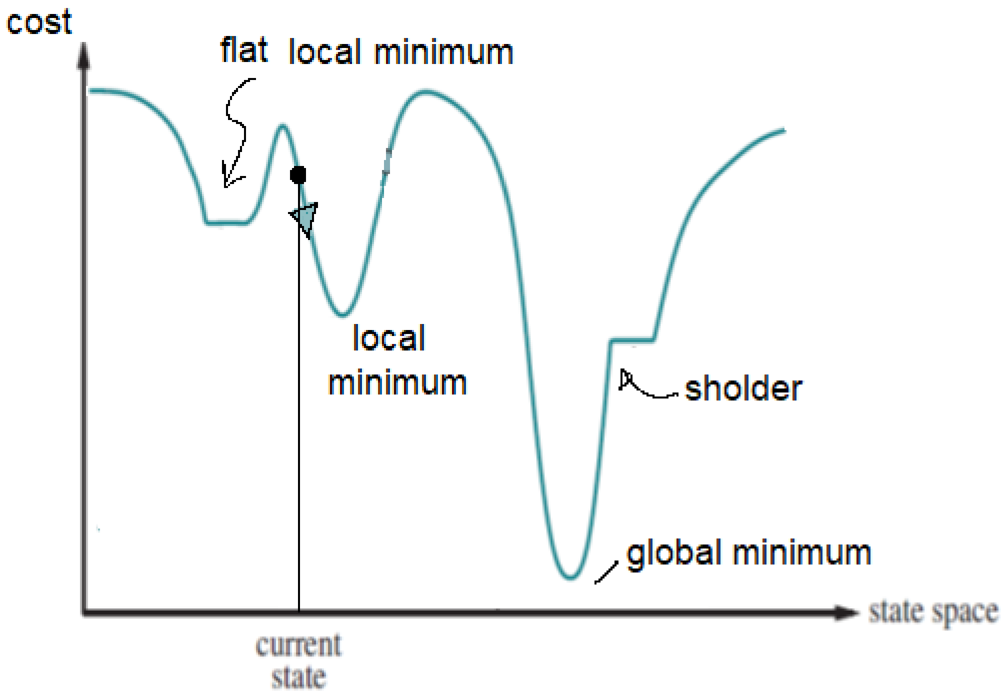
- Use very little memory
- Can often find reasonable solutions in large or infinite (continuous) state spaces.

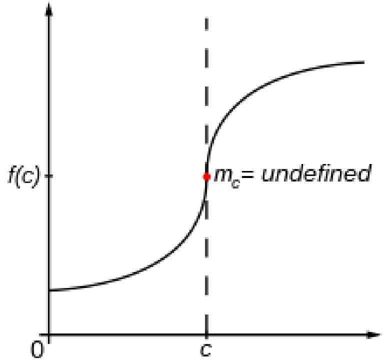
Example



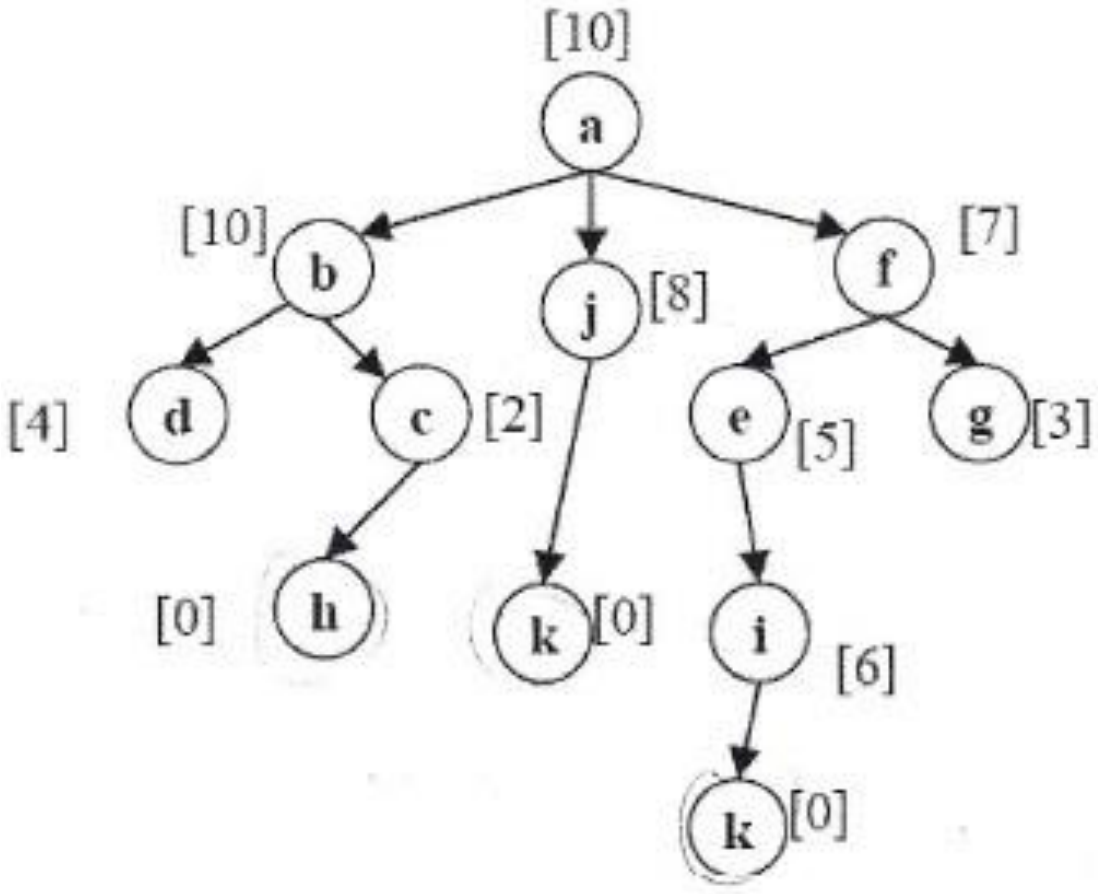
Definitions

gradient descent (also often called **steepest descent**) is a first-order iterative optimization algorithm for finding a local minimum of a differentiable function.





Example



Variants Of Hill Climbing

1. **Stochastic hill climbing** :chooses at Stochastic hill climbing random from among the uphill moves;
2. **First-choice hill climbing :** implements stochastic First-choice hill climbing hill climbing by generating successors randomly until one is generated that is better than the current state.
   - For each restart: run until termination vs. run for a fixed time
   - Run a fixed number of restarts or run indefinitely
3. **Random-restart Hill Climbing**,: If at first you Random-restart hill climbing don't succeed, try, try again." It conducts a series of hill-climbing searches from randomly generated initial states, until a goal is found.

Simulated Annealing

Combine hill climbing with a random walk in a way that yields both efficiency and completeness.

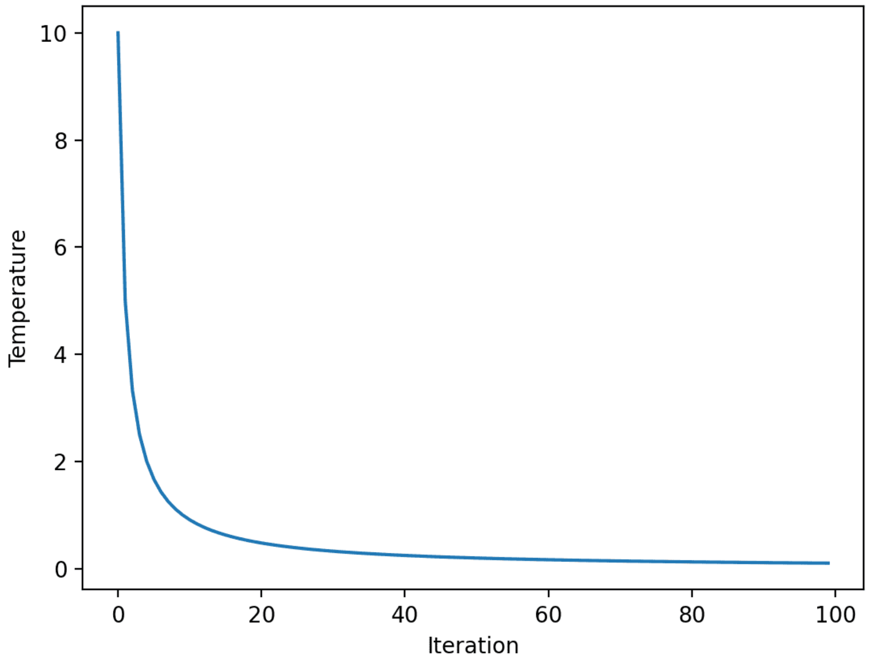
```
function SIMULATED-ANNEALING(problem, schedule) returns a solution state
    current ← problem.INITIAL
    for t = 1 to ∞ do
        T ← schedule(t)
        if T = 0 then return current
        next ← a randomly selected successor of current
        ΔE ← VALUE(current) – VALUE(next)
        if ΔE > 0 then current ← next
        else current ← next only with probability e^(ΔE/T)
```

Instead of picking the best move, however, it picks a random move

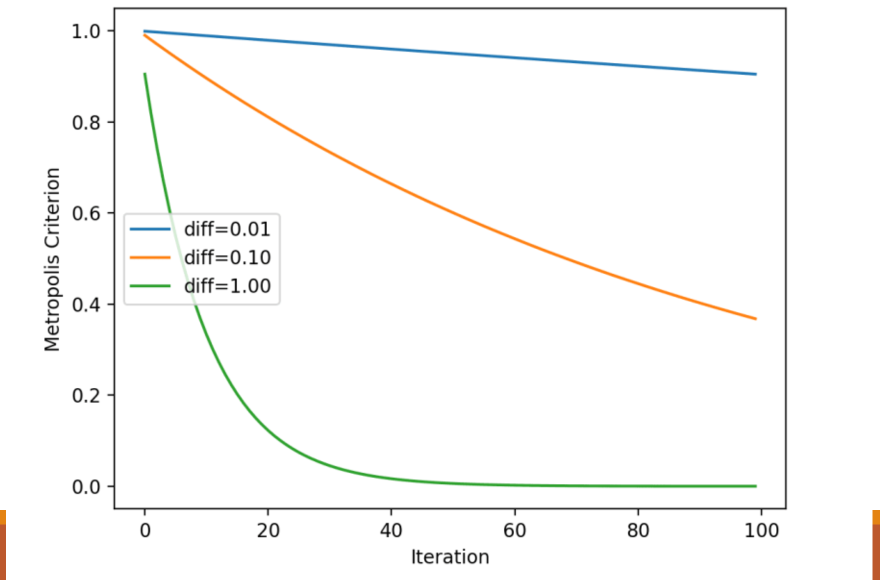
Boltzmann distribution

Simulated Annealing

if $\Delta E > 0$ **then** *current* ← *next*
else *current* ← *next* only with probability $e^{\Delta E/T}$



high T: probability of "locally bad" move is higher
low T: probability of "locally bad" move is lower



Simulated Annealing (Time Gradiant aware)

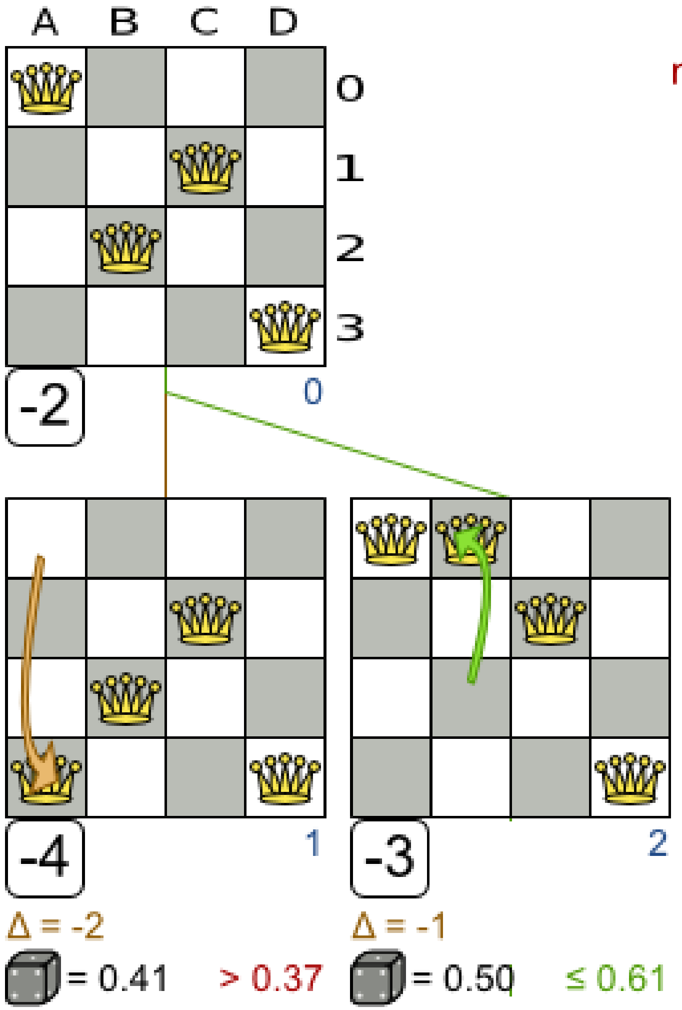
N queens (n = 4, startingTemperature = 2)

Temperature decreases for each step

max $= e^{\Delta/t}$

Step 0

t	Δ	max
2.0	≥0	any
	-1	0.61
	-2	0.37
	-3	0.22
	-4	0.14



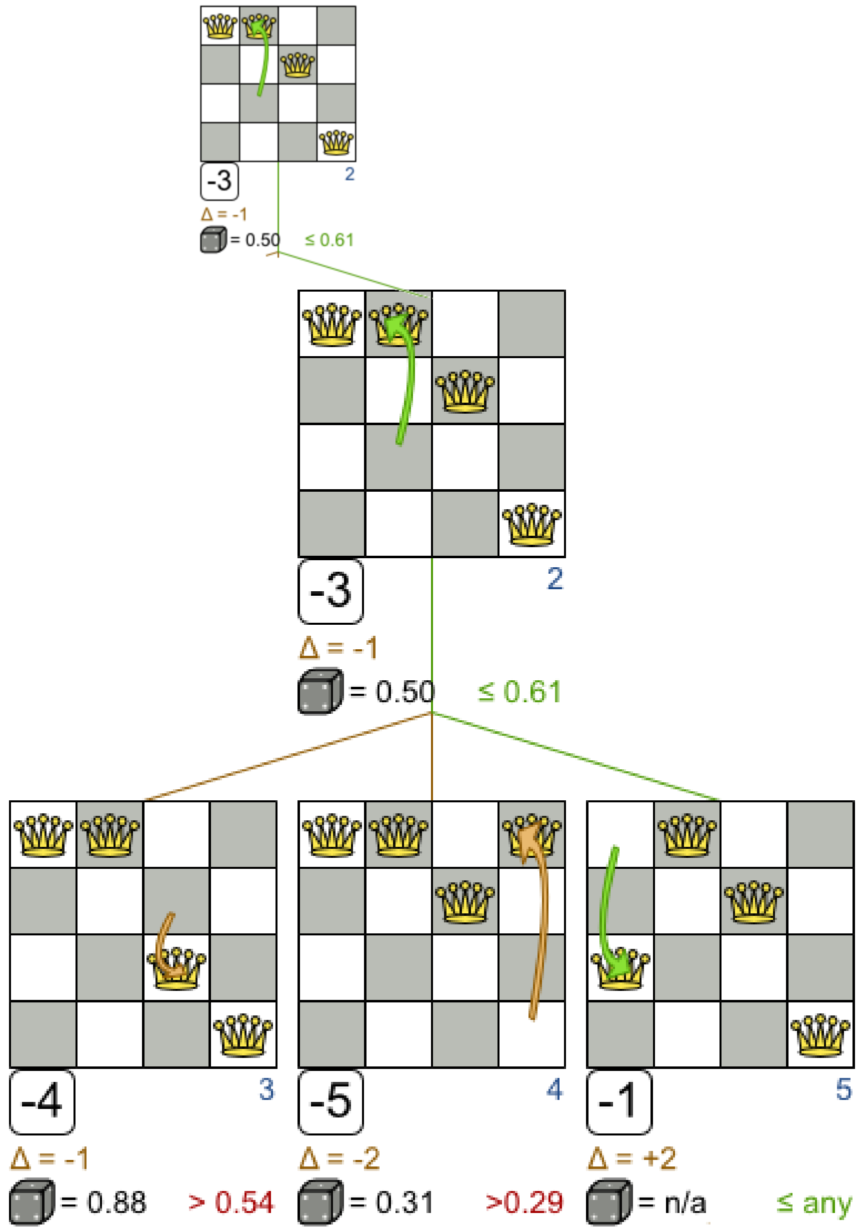
Simulated Annealing (Time Gradiant aware)

N queens (n = 4, startingTemperature = 2)

Step 1

t
1.6

Δ	max 🎲
≥0	any
-1	0.54
-2	0.29
-3	0.15
-4	0.08



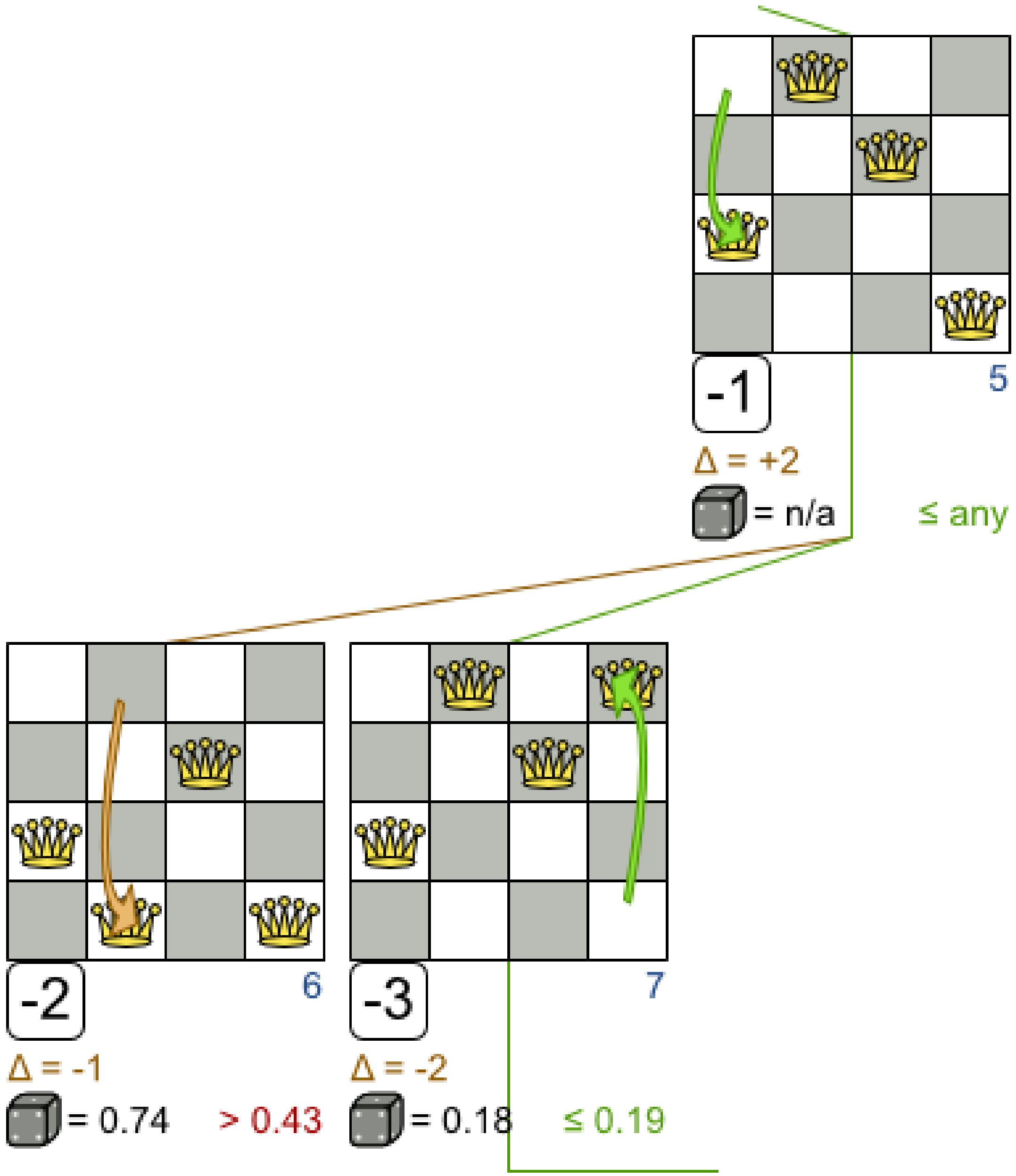
Simulated Annealing (Time Gradiant aware)

N queens (n = 4, startingTemperature = 2)

Step 2

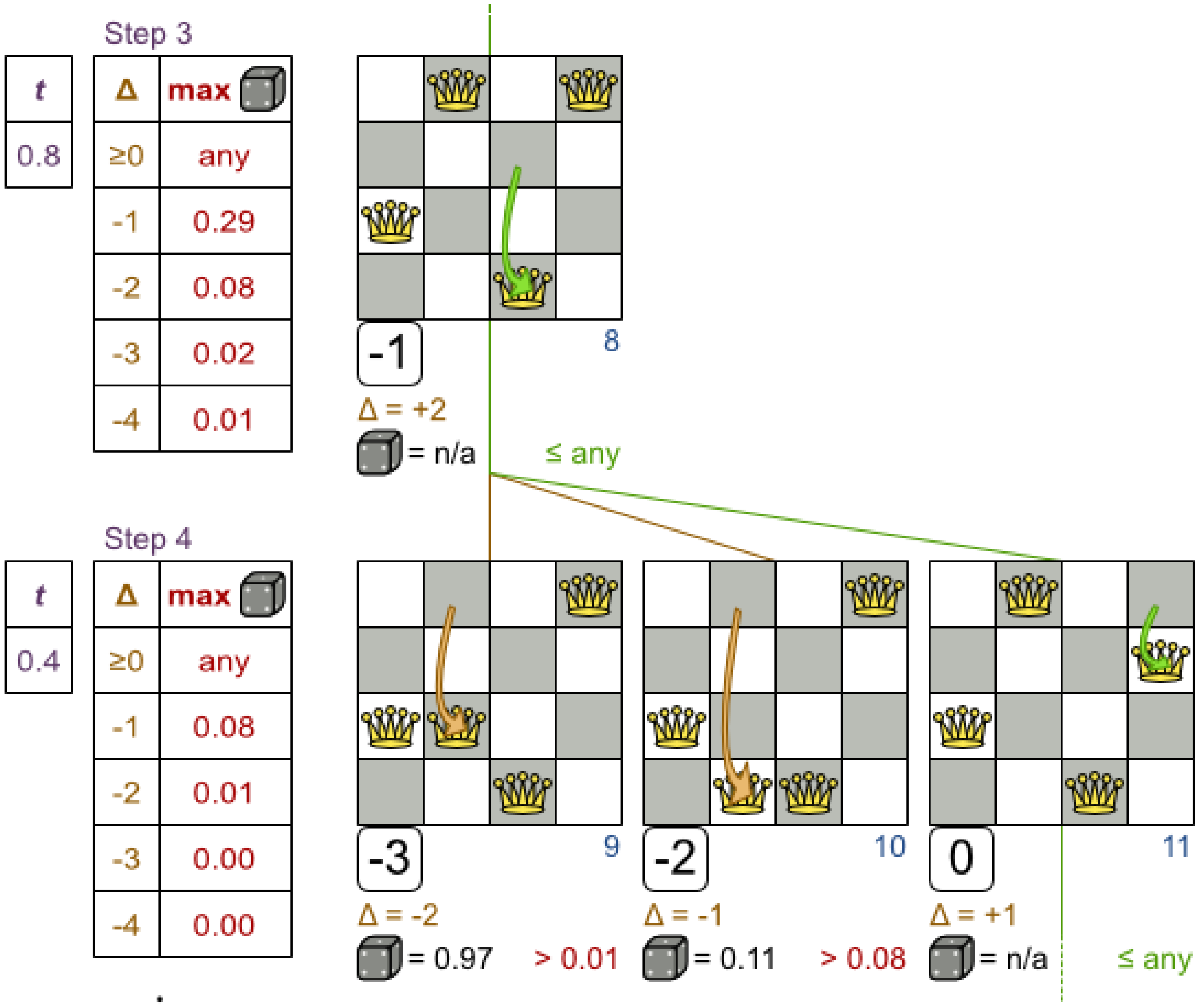
t
1.2

Δ	max
≥0	any
-1	0.43
-2	0.19
-3	0.08
-4	0.04



Simulated Annealing (Time Gradiant aware)

N queens (n = 4, startingTemperature = 2)



Local beam search

Idea: Keeping only one node in memory is an extreme reaction to memory problems.

Local beam save n nodes in stack: k = 1 → Hill climbing , k= ∞ → Best first search

variant called Stochastic Beam Search

Example