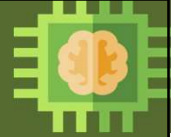


Elective Course

Course Code: CS4103

Autumn 2025-26

**Lecture #11**

Artificial Intelligence for Data Science

Week-3: PROBLEM SOLVING BY SEARCH

Introduction to Informed Search [Part-III]

(Local Search: Hill Climbing and Simulated Annealing)

Course Instructor:**Dr. Monidipa Das**

Assistant Professor

Department of Computational and Data Sciences

Indian Institute of Science Education and Research Kolkata, India 741246

Devising Heuristic Functions



- Relaxation of formally described problems
- Pattern databases
- Learning

Generating Admissible Heuristics



- A **relaxed problem** is a version of a search problem with less restrictions on the applicability of the next-state operators
- **Example: n-puzzle**
 - original: "A tile can move from position p to position q if p is adjacent to q and q is empty"
 - relaxed-1: "A tile can move from p to q if p is adjacent to q "
 - relaxed-2: "A tile can move from p to q if q is empty"
 - relaxed-3: "A tile can move from p to q "
- The exact solution cost of a relaxed problem is often a good (admissible) heuristics for the original problem
- **Key point:** the optimal solution cost of the relaxed problem is no greater than the optimal solution cost of the original problem

Dr. Monidipa Das, Department of CDS, IISER Kolkata

Home Assignment



Consider the input files (*coordinates.txt*, and *graph_data.csv*) containing the details of coordinates of the cities and the road connectivity among them, respectively.

Write a menu-driven Python code that gives users options, such as **Greedy Best First Search**, **UCS**, and **A* search**, to **find a route as well as cost** for reaching any destination city from a given source city. Both the cities should be taken as inputs from the user.

The program should also show the visual representation of the city road network along with indication of source (in text), destination (in text), and the highlighted path as suggested by the selected algorithm.

Node	X	Y
C1	1	9
C2	3	7
C3	2	4
C4	1	1
C5	6	8
C6	6	5
C7	5	3
C8	5	1
C9	9	3
C10	9	7

coordinates.txt

Note:

- Regarding file reading and visualization, you may please take help from Lecture#09

Source	Destination	Cost
C1	C2	4
C2	C1	4
C2	C3	4
C2	C5	4
C3	C2	4
C3	C6	5
C4	C7	11
C4	C8	5
C5	C2	4
C5	C6	9
C6	C3	5
C6	C5	9
C6	C7	8
C6	C9	5
C7	C4	11
C7	C6	8
C7	C8	2
C8	C4	5
C8	C7	2
C8	C9	7
C9	C6	5
C9	C8	7
C9	C10	4
C10	C9	4

graph_data.csv

To be done by a group of **maximum four (04)** students;

Submission:

Submit by **25-AUG-2025 11:59 PM**

(through email with subject "CS4103: Assignment on Road Network" and cc to your group-mates, if there is any)

Please write the code yourself. DO NOT use AI tools to develop the code.

Dr. Monidipa Das, Department of CDS, IISER Kolkata

Desired Output



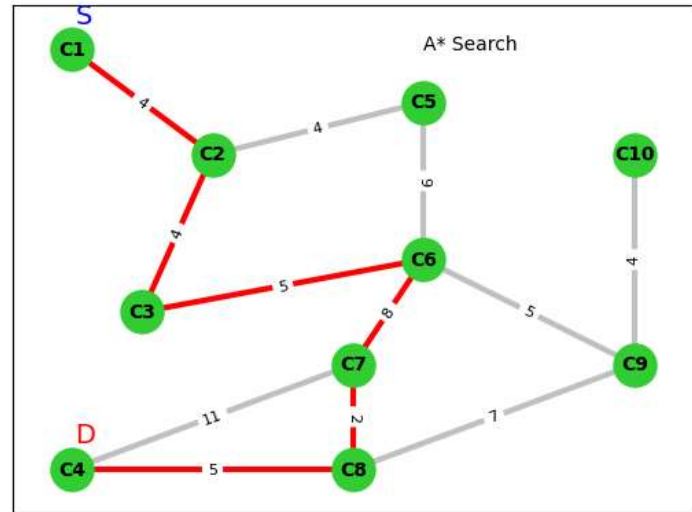
```
=====
OPTIONS AVAILABLE:
=====
1. Uniform Cost Search (UCS)
2. Greedy Best First Search
3. A* Search
4. Exit

Please select an option from above: 3
See the road network and enter the source city:C1
See the road network and enter the destination city:C4
You have selected A* Search. Following is the suggested path:

Path: C1-->C2-->C3-->C6-->C7-->C8-->C4
Path Cost: 28

=====
OPTIONS AVAILABLE:
=====
1. Uniform Cost Search (UCS)
2. Greedy Best First Search
3. A* Search
4. Exit

Please select an option from above: 4
Thank you!....Good bye.....
```



Dr. Monidipa Das, Department of CDS, IISER Kolkata



Beyond Classical Search

Local Search: Hill Climbing

Dr. Monidipa Das, Department of CDS, IISER Kolkata

Local Search



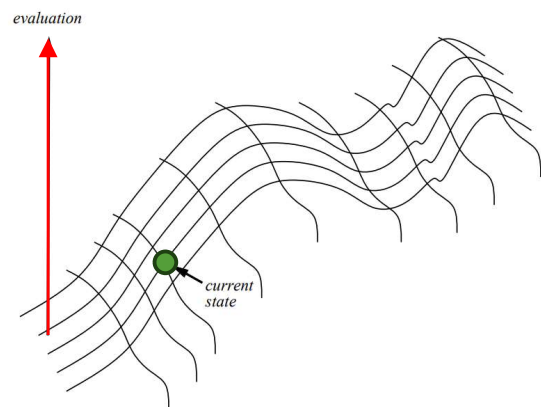
- In many optimization problems the goal state itself is the solution
- The state space is a set of complete configurations
- Search is about finding the *optimal* configuration or just a *feasible* configuration
- In such cases, one can use *iterative improvement*, or *local search*, methods
- An evaluation, or *objective*, function h must be available that measures the *quality* of each state
- **Main Idea:** Start with a random initial configuration and make small, local changes to it that improve its quality

Dr. Monidipa Das, Department of CDS, IISER Kolkata

Local Search: The Landscape Metaphor



- Ideally, the evaluation function h should be *monotonic*: the closer a state to an optimal goal state the better its h -value.
- Each state can be seen as a point on a surface.
- The search consists in moving on the surface, looking for its highest peaks (or, lowest valleys): the optimal solutions.

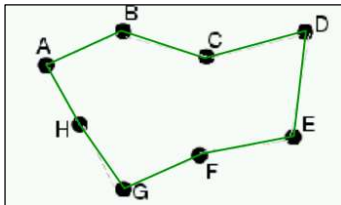


Dr. Monidipa Das, Department of CDS, IISER Kolkata

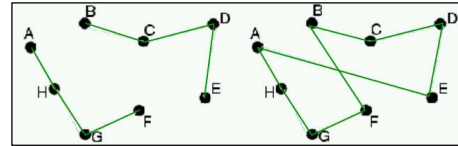
Local Search Example: TSP



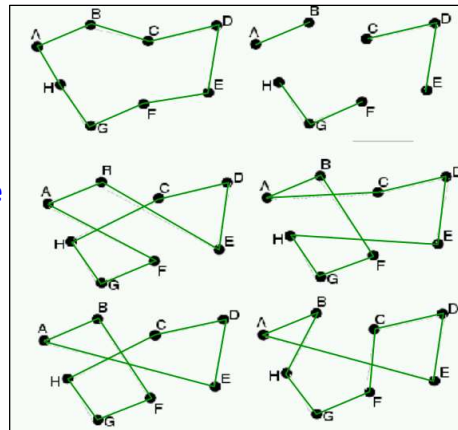
- **TSP:** Travelling Salesperson Problem
- h = length of the tour
- **Strategy:** Start with any complete tour, perform pairwise exchanges



2-Change Example



3-Change Example



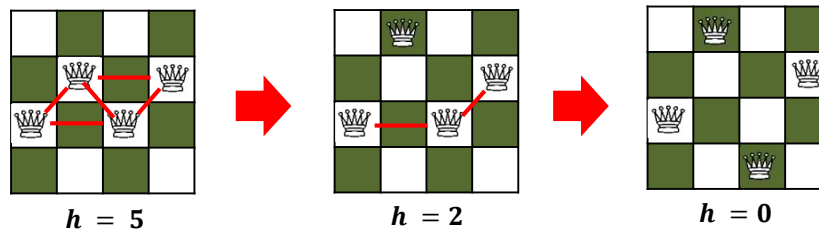
- Variants of this approach get within 1% of optimal very quickly with thousands of cities

Dr. Monidipa Das, Department of CDS, IISER Kolkata

Local Search Example: n -queens



- Put n queens on an $n \times n$ board with no two queens on the same row, column, or diagonal
- h = number of conflicts
- **Strategy:** Move a queen to another square in the same column



- Almost always solves n -queens problems almost instantaneously for very large n , e.g., $n = 10^6$

Dr. Monidipa Das, Department of CDS, IISER Kolkata

Local Search: Boolean Satisfiability or SAT Problem



- **SAT:** determining if a Boolean formula is satisfiable or unsatisfiable.
 - **Satisfiable** : If the Boolean variables can be assigned values such that the formula turns out to be TRUE
- ***h*** = number of satisfied clauses
- **Current state:** (A=T, B=F, C=T, D=T, E=T)
- **Strategy:** flip the assignment of one variable

(A=**F**, B=F, C=T, D=T, E=T)
 (A=T, B=**T**, C=T, D=T, E=T)
 (A=T, B=F, C=**F**, D=T, E=T)
 (A=T, B=F, C=T, D=**F**, E=T)
 (A=T, B=F, C=T, D=T, E=**F**)

$A \vee \neg B \vee C$
 $\neg A \vee C \vee D$
 $B \vee D \vee \neg E$
 $\neg C \vee \neg D \vee \neg E$
 $\neg A \vee \neg C \vee E$

Dr. Monidipa Das, Department of CDS, IISER Kolkata

Hill-Climbing Search



- “Like climbing Everest in thick fog with amnesia”

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

inputs: *problem*, a problem

local vars: *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*]

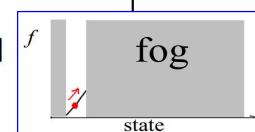
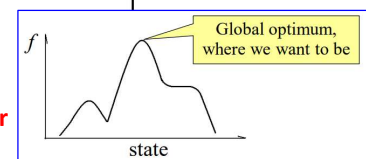
current ← *neighbor*

end

Best Successor

worse than

Steepest Ascent/Descent version

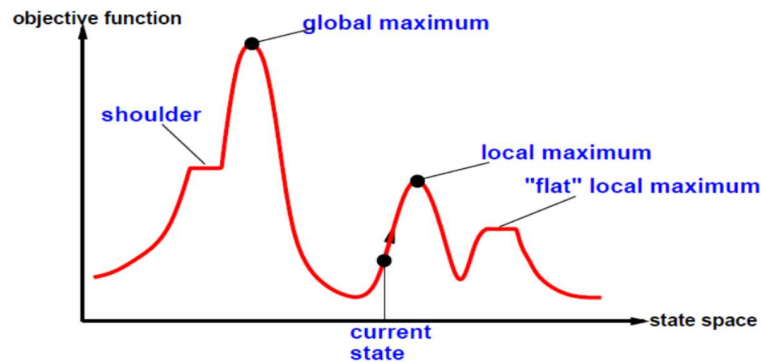


Dr. Monidipa Das, Department of CDS, IISER Kolkata

Hill Climbing in 1-D State-space Landscape



- Following is a one-dimensional state-space landscape in which elevation corresponds to the objective function.



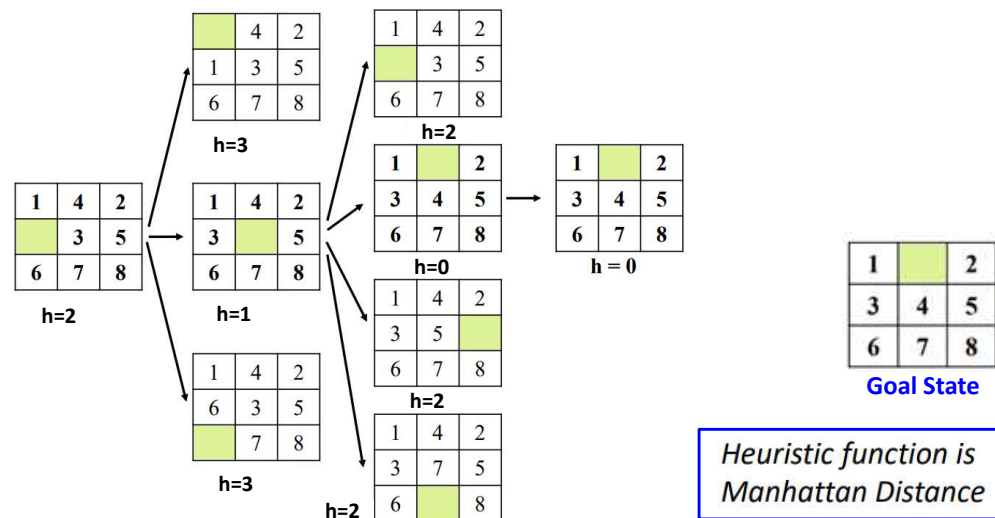
- The aim is to find the global maximum. Hill-climbing search modifies the current state to try to improve it, as shown by the arrow

Dr. Monidipa Das, Department of CDS, IISER Kolkata

Hill Climbing Example



- 8-puzzle**



Dr. Monidipa Das, Department of CDS, IISER Kolkata

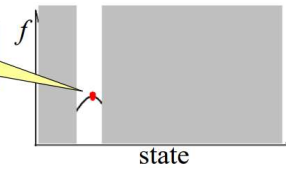
Hill Climbing: Disadvantages



- Local Maximum**

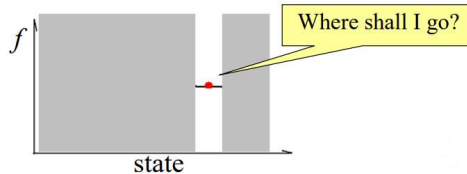
- A state that is better than all of its neighbors, but not better than some other states far away

Declare top-of-the-world?



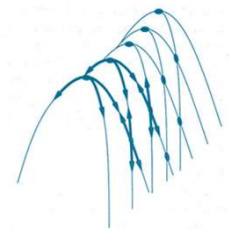
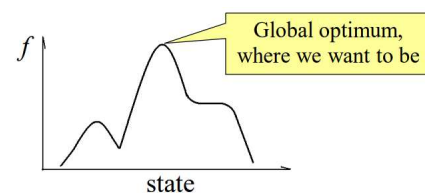
- Plateau**

- A flat area of the search space in which all neighboring states have the same value



- Ridge**

- Ridges result in a sequence of local maxima that is very difficult for greedy algorithms to navigate.

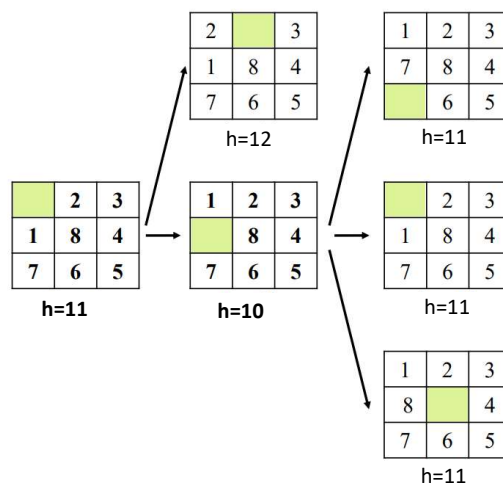


Dr. Monidipa Das, Department of CDS, IISER Kolkata

Hill Climbing Disadvantage: Example



- 8-puzzle: stuck at local optimum**



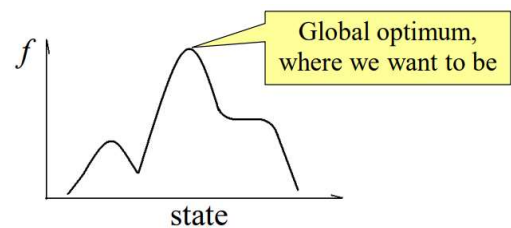
Dr. Monidipa Das, Department of CDS, IISER Kolkata

Hill Climbing: Disadvantages



- **Ways Out**

- Random selection of neighbor
- Make a **big jump** to try to get in a new section.
- Move in **several directions** at once



Dr. Monidipa Das, Department of CDS, IISER Kolkata

First-Choice Hill Climbing



- **First-choice hill climbing** implements stochastic hill climbing by generating successors randomly **until one is generated that is better than the current state**.
- This is a good strategy **when a state has many successors**.
- First-choice hill climbing is also **NOT complete**

Dr. Monidipa Das, Department of CDS, IISER Kolkata

Stochastic Hill Climbing



- **Stochastic hill climbing** chooses at random from among the uphill moves;
- The **probability of selection** can vary with the **steepness of the uphill move**.
- Stochastic hill climbing usually **converges more slowly** than steepest ascent, **but in some state landscapes, it finds better solutions**.
- Stochastic hill climbing is **NOT complete**, but it may be **less likely to get stuck**.

Dr. Monidipa Das, Department of CDS, IISER Kolkata

Random-Restart Hill Climbing



- **Random-Restart Hill Climbing** conducts a **series of hill-climbing searches from randomly generated initial states, until a goal is found**.
- Random-Restart Hill Climbing is **complete if infinite (or sufficiently many) tries are allowed**.
- If each hill-climbing search has a probability p of success, then the expected number of restarts required is $1/p$.
- **Success of hill climbing depends very much on the shape of the state-space landscape:**
 - If there are few local maxima and plateau, random-restart hill climbing will find a good solution very quickly.
 - On the other hand, many real problems have many local maxima to get stuck on.
 - NP-hard problems typically have an exponential number of local maxima to get stuck on.

Dr. Monidipa Das, Department of CDS, IISER Kolkata



Simulated Annealing

Another Local Search Technique

Dr. Monidipa Das, Department of CDS, IISER Kolkata



Annealing Process

- Annealing is a thermal process for obtaining low energy states of a solid in a heat bath.
- **Annealing Process**
 - **Raising the temperature up** to a very high level (melting temperature, for example), the atoms have a higher energy state and a high possibility to re-arrange the crystalline structure.
 - **Cooling down slowly**, the atoms have a lower and lower energy state and a smaller and smaller possibility to re-arrange the crystalline structure.

Dr. Monidipa Das, Department of CDS, IISER Kolkata

Analogy



- **Analogy –**
 - Metal \leftrightarrow Problem
 - Energy State \leftrightarrow Cost Function
 - Temperature \leftrightarrow Control Parameter
 - A completely ordered crystalline structure \leftrightarrow the optimal solution for the problem

Global optimal solution can be achieved as long as the cooling process is slow enough.

Dr. Monidipa Das, Department of CDS, IISER Kolkata

Simulated Annealing



- **Based on a metallurgical metaphor**
 - Start with a temperature set very high and slowly reduce it.
 - Run hill climbing with the twist that you can occasionally replace the current state with a **worse state** based on the current temperature and how much worse the new state is.
- **More formally...**
 - Generate a random new neighbor from current state.
 - **If it is better take it.**
 - If it is worse, then take it with **some probability proportional to the temperature and the delta between the new and old states.**

Dr. Monidipa Das, Department of CDS, IISER Kolkata

Simulated Annealing



function SIMULATED-ANNEALING(*problem*, *schedule*) **returns** a solution state

inputs: *problem*, a problem

schedule, a mapping from time to “temperature”

current \leftarrow MAKE-NODE(*problem*.INITIAL-STATE)

for $t = 1$ **to** ∞ **do**

...to be continued in the next class

next \leftarrow a randomly selected successor of *current*

$\Delta E \leftarrow next.VALUE - current.VALUE$

if $\Delta E > 0$ **then** *current* \leftarrow *next*

else *current* \leftarrow *next* only with probability $e^{\Delta E/T}$

Dr. Monidipa Das, Department of CDS, IISER Kolkata



Questions?

Dr. Monidipa Das, Department of CDS, IISER Kolkata