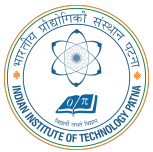
CS5201: Advanced Artificial Intelligence

Local search

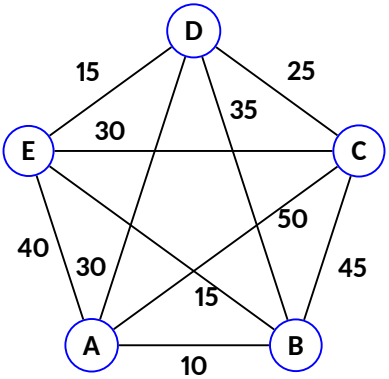


Arijit Mondal
Dept of Computer Science and Engineering
Indian Institute of Technology Patna
www.iitp.ac.in/~arijit/

Local search

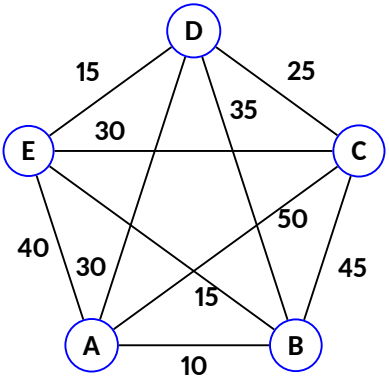
- Search based methods systematically explore all paths in the state space
- For certain problems, the path in which it is explored is not important
- The final configuration is more important
  - Example: N-queens, travelling salesperson problem, vehicle routing problem, etc.
- If the path does not matter, we can apply local search method
  - It operates on a single node and generally moves to a neighbor node
- Advantage of local search
  - Low memory overhead
  - Can find reasonable solution in a large / infinite state-space for which a systematic search is unsuitable
- Local search based methods are more suitable for optimization problems

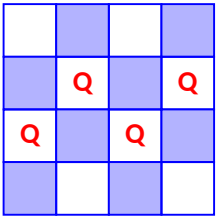
Example



[ABCDEA], 135
[ABCEDA], 130

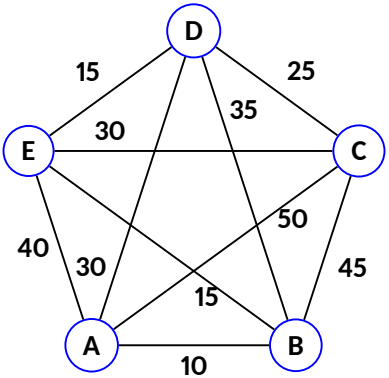
Example

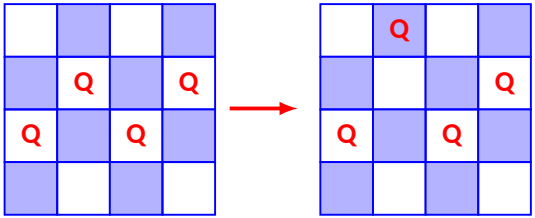




[ABCDEA], 135
[ABCEDA], 130

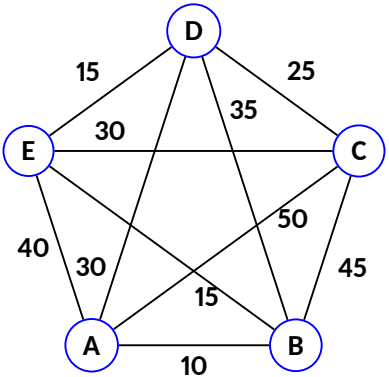
Example



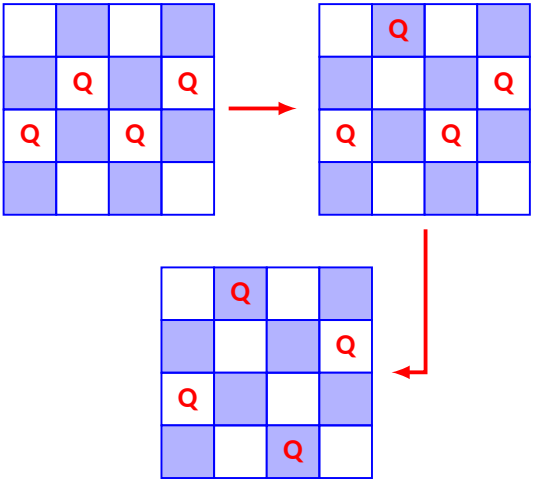


[ABCDEA], 135
[ABCEDA], 130

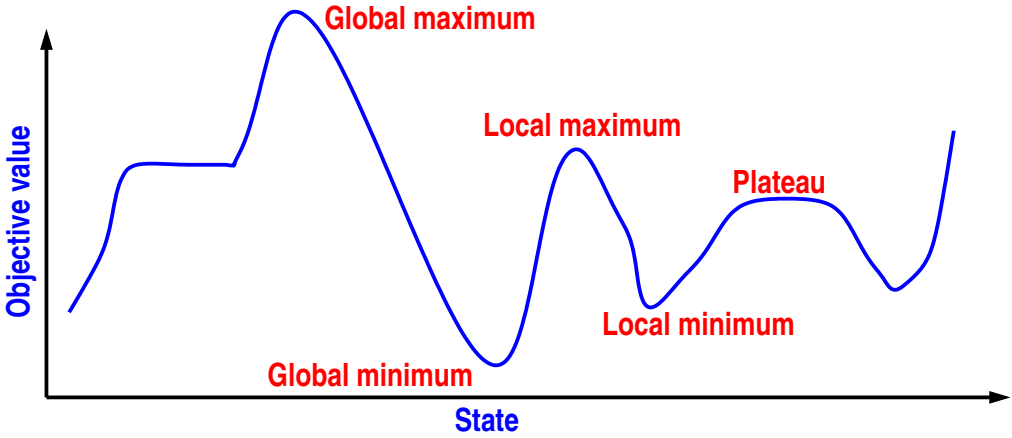
Example



[ABCDEA], 135
[ABCEDA], 130



Cost curve

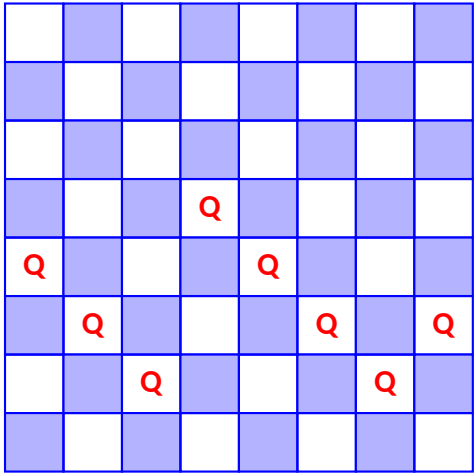

Global maximum
Local maximum
Plateau
Local minimum
Global minimum
Objective value
State

Hill climbing (steepest ascent)

- **Steps:**
  1. current = initial configuration
  2. loop do
  3. neighbor = highest valued nodes from the successor of current
  4. if (neighbor.value $\leq$ current.value) then exit
  5. current = neighbor
  6. end
- **Above steps are for maximization problem. In a similar manner, minimization problem can be solved**

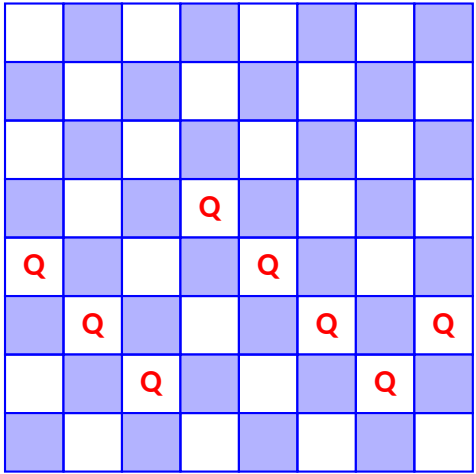
Hill climbing: 8-queens

- Consider 8-queens problem
- Heuristic measure (h): number of pairs attacking each other directly or indirectly
- Number of successor states is 56



Hill climbing: 8-queens

- Consider 8-queens problem
- Heuristic measure (h): number of pairs attacking each other directly or indirectly
- Number of successor states is 56



$h = 17$

Hill climbing: 8-queens

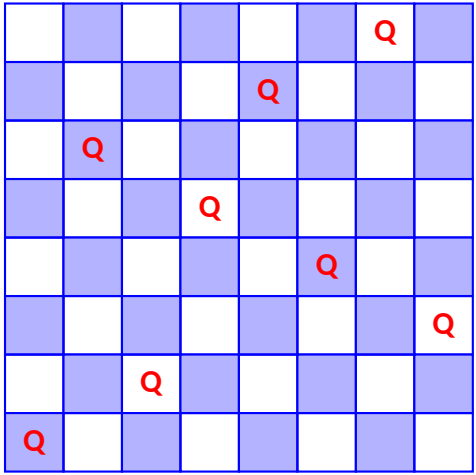
- Consider 8-queens problem
- Heuristic measure (h): number of pairs attacking each other directly or indirectly
- Number of successor states is 56

18	12	14	13	13	12	14	14
14	16	13	15	12	14	12	16
14	12	18	13	15	12	14	14
15	14	14	Q	13	16	13	16
Q	14	17	15	Q	14	16	16
17	Q	16	18	15	Q	15	Q
18	14	Q	15	15	14	Q	16
14	14	13	17	12	14	12	18

$h = 17$

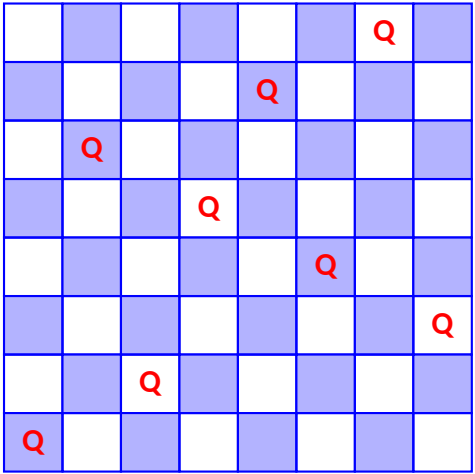
Hill climbing: 8-queens

- Consider 8-queens problem
- Heuristic measure (h): number of pairs attacking each other directly or indirectly



Hill climbing: 8-queens

- Consider 8-queens problem
- Heuristic measure (h): number of pairs attacking each other directly or indirectly



$h = 1$

Hill climbing: 8-queens

- Consider 8-queens problem
- Heuristic measure (h): number of pairs attacking each other directly or indirectly

3	3	3	3	2	3	Q	3
3	3	4	2	Q	4	2	4
2	Q	3	3	5	4	2	3
3	2	4	Q	4	4	3	2
3	3	4	3	4	Q	2	3
3	5	3	2	4	3	2	Q
4	3	Q	2	2	3	3	3
Q	3	3	2	2	3	2	3

$h = 1$

Variants of Hill climbing

- Sideways move: if no upward move is possible, allow to move a state having the same value
- Stochastic hill climbing: selection among the available uphill moves is done randomly
- First choice hill climbing: successors are generated randomly, one at a time, until one that is better than the current state is found
- Random restart: start randomly from a new position when stuck

Simulated annealing

- This is inspired from physics
- In metallurgy, annealing is used to harden the metal
- Typically, the metal is heated and then allowed to cool down.
- The process repeats for a number of times
- Similar idea is used for optimization

Simulated annealing

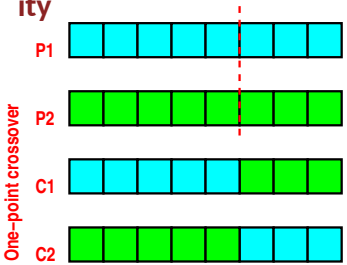
1. current = initial state, it can be generated randomly
2. for t=1 to ∞ do
3. T = schedule(t)
4. if T=0 return current
5. neighbor = randomly selected successor of current
6. ΔE = neighbor.value – current.value
7. if $\Delta E > 0$ then current = neighbor
8. else current = neighbor only with probability exp($\Delta E/T$)

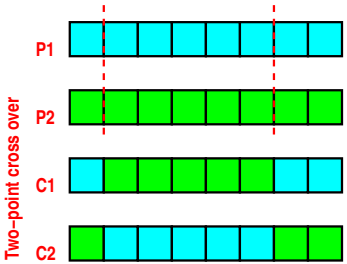
Beam search

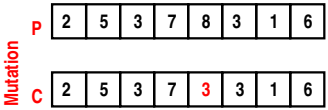
- Maintain k states rather than just one. Begin with k randomly generated states
- In each iteration, generate all the successors of all k states
- Stop if a goal state is found; otherwise Select the k best successors from the complete list and repeat

Genetic algorithms

- This is inspired from evolution of biology
- States are strings over finite alphabet (chromosome)
- Begin with k randomly generated states / chromosomes (population)
- Select individual for next generation based on fitness functon
- Two types of oeprators are used to generate next states:
  - Crossover: fit parents to yield next generation (offspring)
  - Mutation: mutate a parent to create an offspring randomly with some low probability







Thank you!