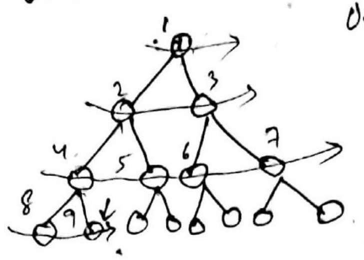
19/08/19

AI

Completeness: Will the algo be able to find out a solution if one exists.

BFS Order: 1 2 3 4 5 6 7 8 9



Path: 1 2 4 9

DFS Order: 1 2 4 8 9

Path: 1 2 4 9

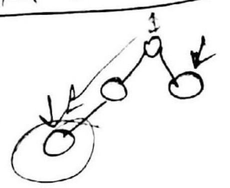
Time Complexity: No. of nodes examined before reaching goal node. $O(b^d)$ → depth of goal node; b ← branching factor

Space complexity: memory required. DFS → $O(n)$, BFS → $O(b^d)$

Optimal solution: Will the algo give an optimal solution

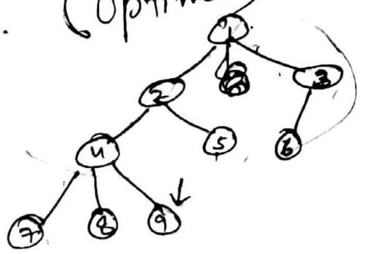
BFS will always provide optimal.

DFS is not complete not optimal



Depth limited search / depth bound search: specify the max depth to search.

Iterative deepening search: Increase max depth. (Optimal). (complete) Time: $O(b^d)$ space: $O(n)$



Iteration	
0	1
1	1 2 3
2	1 2 4 5 3 6
3	1 2 4 7 8 9

Iterative broadening search: $O(b^d)$ time

Not optimal

Not complete

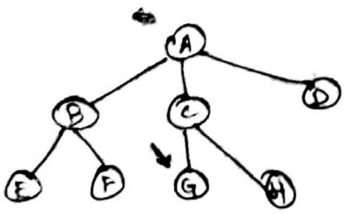
Iteration	
0	1
1	1 2 4 7 5 3 6
2	1 2 4 7 8
3	1 2 4 7 8 9

When will IB be better than ID?

Island driven search.

Graph search algo.

1. Create a search tree Tr. consisting solely of the start node No. Put No on an ordered list called 'open.'
2. Create a list called 'closed' that is initially empty.
3. If 'open' is empty exit with failure
4. Select the first node on 'open' remove it from 'open' and put it on 'closed'. call this node N.
5. If N is a goal node exit successfully with the solution obtained by tracing a path backward along the arcs in Tr. from N to No(arcs are created in step 6)
6. Expand node N, generating a set M of successors. Install M as successors of N in Tr by creating arcs from N to each member of M. Successor will be added to 'open.'
7. Reorder the list 'open' either according to some arbitrary scheme or according to heuristic merit.
8. Goto step 3.



open: A
closed:

Tr: A

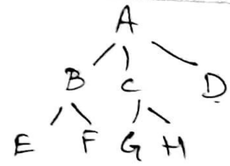
open: B C D.
closed: A

Tr :

open: C D E F
closed: A B.

open: D E F G H.
closed: A B C

Tr:



open: D E F G H.
closed: A B C D.

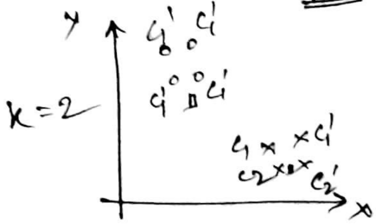
open: G H ← how many not yet expanded
closed: A B C D E F ← order of nodes visited.

open: H.
closed: A B C D E F G ← done goal node.

Informed search: Info is given on the node based on some heuristic.

22/08/19

PR



Exhaustive search ← guaranteed optimal solution (best solution)

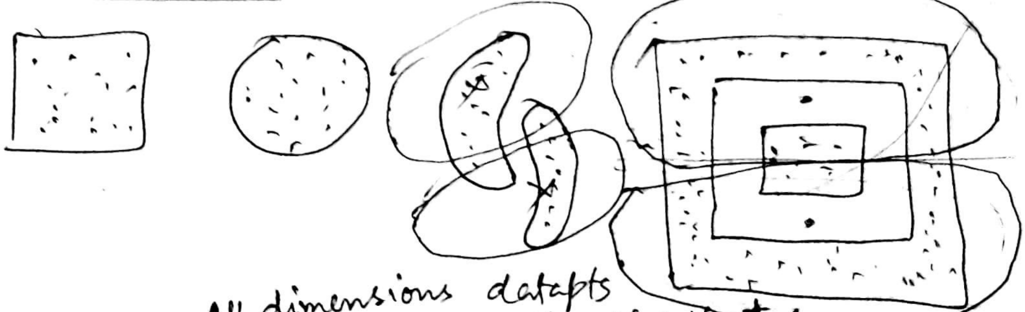
$S(56, 3) > 10^{18}$

So heuristics.

K-means.

1. Select cluster centers randomly from datapoint. Assign each point to each cluster centroid.
2. Recompute cluster centroids.

K-means fails.



All dimensions datapts are equally distributed.

K-means ↳ spherical clusters. Cannot handle overlapping clusters

04/09/19 Miss

Artificial Intelligence

Informed search

- Add domain specific information to select the path.
- Controls | guides search processes
- Heuristic function ($\hat{h}$) estimates the cost of reaching ~~the~~ a goal node from the present state in the search ~~state~~ space, small values of $\hat{h} \rightarrow$ best node.
- A heuristic is a method that might not always find the best solution but is guaranteed to find a good solution in reasonable time.
- By sacrificing completeness it increases efficiency.

eg:- 8-puzzle : # tiles misplaced ($\hat{h}_1(n)$), Manhattan distance ($\hat{h}_2(n)$)

$\hat{h}_1(n) \leq h(n) \leftarrow$ actual value

$\hat{h}_2(n) \leq h(n)$

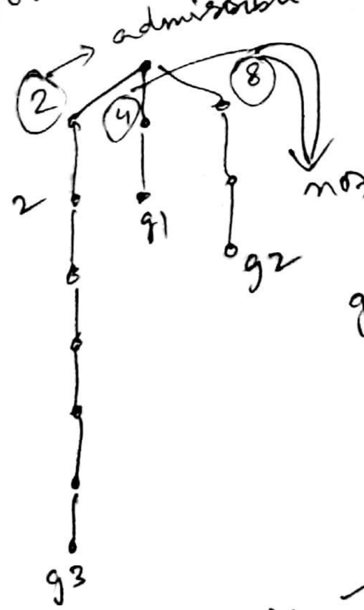
Admissible heuristic. For all nodes n of the graph $\hat{h}(n) \leq h(n)$. (for minimisation problem).

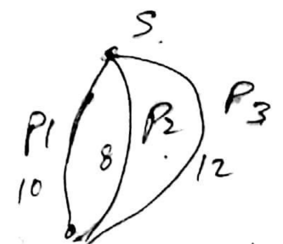
using $\hat{h}(n)$ and best first greedy search algorithm.

$f(n) = \hat{h}(n)$

for A* algorithm $f(n) = g'(n) + \hat{h}(n)$ ← edge cost

$\hat{h}(n)$ must be admissible in nature.





$g'(n) = g(n)$ when all paths are explored from start state to current state.

$g' = 10$

$= 8$

$= 8$

$g' \geq g$ ↓ actual lowest cost path

A* states if $\hat{h}$ is admissible then we will get global minimum.

09/09/19

AI

~~23rd Monday CT~~
26th Thu CT.

IDA*

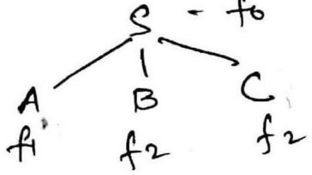
Time: In worst case if A^* expands N nodes, IDA^* expands $O(N^2)$ nodes. → i.e. all f-cost are unique 1, 2, 3, ... n.

Space: $O(b(f^*/\epsilon))$ f^*: optimal cost.

Repeatedly search in DFS fashion over subgraph with f-cost less than ϵ. Less than 2ϵ until a goal node is found (where, $\epsilon <= g(n, m)$ for all $n, m \in S(n)$).

All nodes with same f-cost value will be searched.

$f_0 < f_1 < f_2 < f_3$



It.

1 S

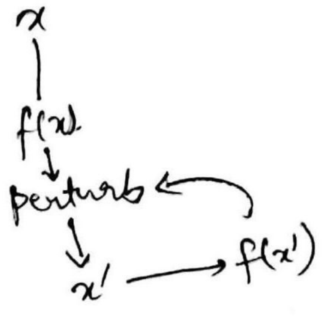
2 S A.

3 S A B C

Iterative Improvement Search

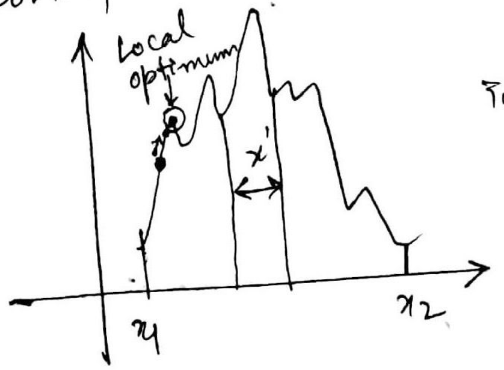
- Every state represents a complete solution to the problem, although not necessarily an optimal one.
- Initialize the algo at some (random) state, iteratively perturb the current solution, update if any of the tested perturbations yield an improvement.
- Never backs up, no search tree.

Hill climbing, Simulated annealing, Genetic Algo.



Hill climbing.

- walk as much uphill as possible.
- move from one point to that adjacent point having highest elevation.
- Soln. found, but not necessarily an optimal one.



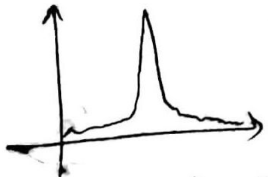
Total: $x_2 - x_1$

Probab of reaching optimal solution.

$$\frac{x'}{x_2 - x_1}.$$

Drawbacks

- Local optimal foothill problem — attracted to local optima.
- Plateau problem — nowhere to turn on the flat.
- Ridge problem — no ~~step~~ available operator to move from the region.



* Use a no. of simulations with different initial value
Iterated hill climb algo.
* For plateau change only for a no. of iterations and if no change stop

Iterated hill climber algo:

1. Begin.
2. t=0
3. Repeat
   4. local = false.

5. Select a random state S_c and calculated its functional value $f(S_c)$.

6. Repeat

7. Select n new solutions in the neighborhood of S_c. (by flipping randomly single bit of S_c and calc. their fn al values)

8. Select the solution S_n from the set of new solutions with largest fn.al value $f(S_n)$.

9. If $f(S_c) < f(S_n)$ set $S_c = S_n$ else local = true.

10. until local.

11. $t = t+1$

12. until $t = max$.

13. end.

Simulated Annealing :

- Simulates the annealing of physical systems.
- In statistics mechanics, annealing is a process of slowly cooling a physical system from a suff. high temp. in order to obtain thermal equilibrium with global minimum energy.
- At any temp T, the system will be in state s with energy value E(s) by following Boltzmann probability distribution.

$$P_T(s) = \frac{e^{-\frac{E(s)}{kT}}}{\sum_{w} e^{-\frac{E(w)}{kT}}}$$

k : Boltzmann constant   w : set of all states correspondence.

T is a parameter.

S: state of a system. — solution

E(S): functional value f(S).

We may get global solution.

Algo works bet if decay of T is logarithmic.

At T → 0 global minimum energy.

- very slow process.

metropolis simulation.

Montecarlo Simulation.

HC, SA ← single solution based search technique.

AI GA ~~miss~~.

16/09/19

AI production system

Shows a separation between data, operation and control.

↓

Select a path among a set of all alternatives.

Rule selection.

4	5	6
2	1	8
	3	7

up, right

Control strategy

- irrevocable.
- tentative.
  - backtracking
  - graph search control.

Procedure : production

1. 'data' is initial database.
2. until 'data' satisfies the termination condition do
   - 2.1. Begin
   - 2.2. Select some rule R in the set of rules that can be applied to 'data'.
   - 2.3. 'data' is the result of applying R to 'data'.

Scanned by CamScanner