Informed Search

HILL CLIMBING SEARCH

1

Tries to improve the efficiency of **depth-first.**

- *Informed depth-first algorithm.*

2

An iterative algorithm that starts with an ***arbitrary solution*** to a problem, then attempts to find a better solution by incrementally changing a single element of the solution.

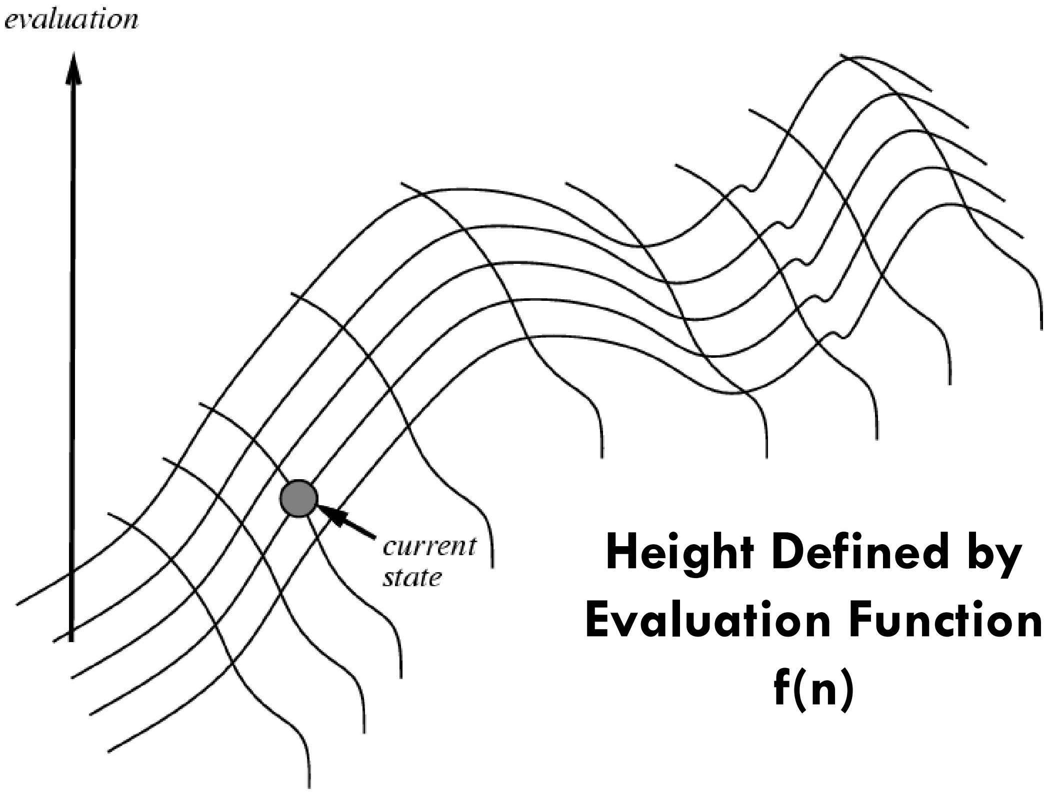
3

It **sorts** the successors of a node (according to their heuristic values) before adding them to the list to be expanded.

4

If the change produces a better solution, an incremental change is made to the new solution, repeating until no further improvements can be found.

HILL CLIMBING ON A SURFACE OF STATES



Height Defined by Evaluation Function f(n)

HILL CLIMBING SEARCH

Looks one step ahead to determine if any successor is better than the current state; if there is, move to the best successor.

Rule:

- If there exists a successor s for the current state n such that
  - $h(s) < h(n)$ and
  - $h(s) \leq h(t)$ for all the successors t of n,
- Then move from n to s. Otherwise, halt at n.

HILL CLIMBING SEARCH

"Like climbing Everest in thick fog with amnesia"

Hill Climbing VS. Greedy search?

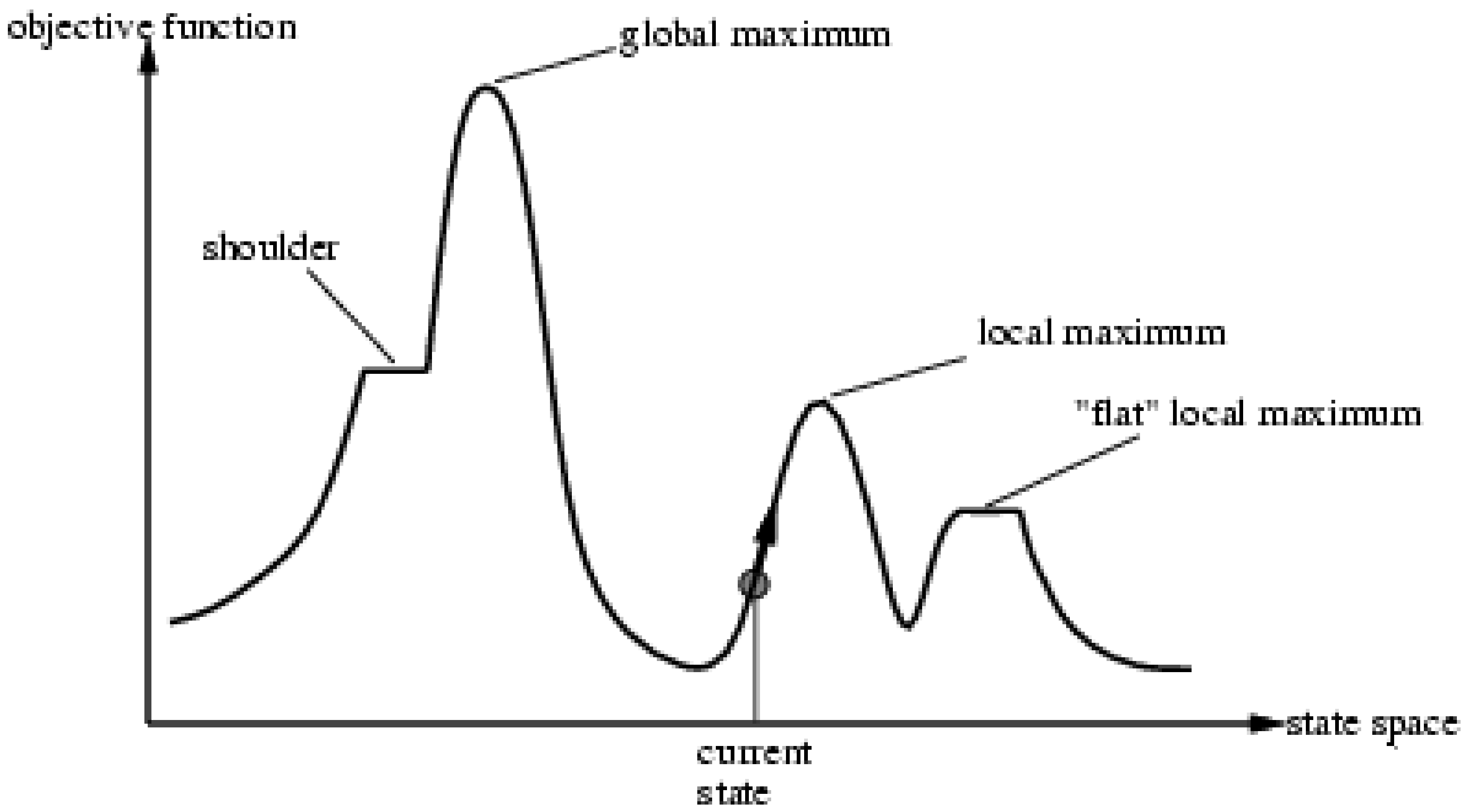
Similar to Greedy search in the sense that it uses *h*(), but ***does not allow backtracking*** or ***jumping to an alternative path*** since it doesn't "**remember**" where it has been.

Hill Climbing VS. Beam search

Corresponds to Beam search with a beam width of 1 (i.e., the maximum size of the nodes list is 1).

Not complete since the search will terminate at "local minima," "plateaus," and "ridges."

HILL CLIMBING SEARCH

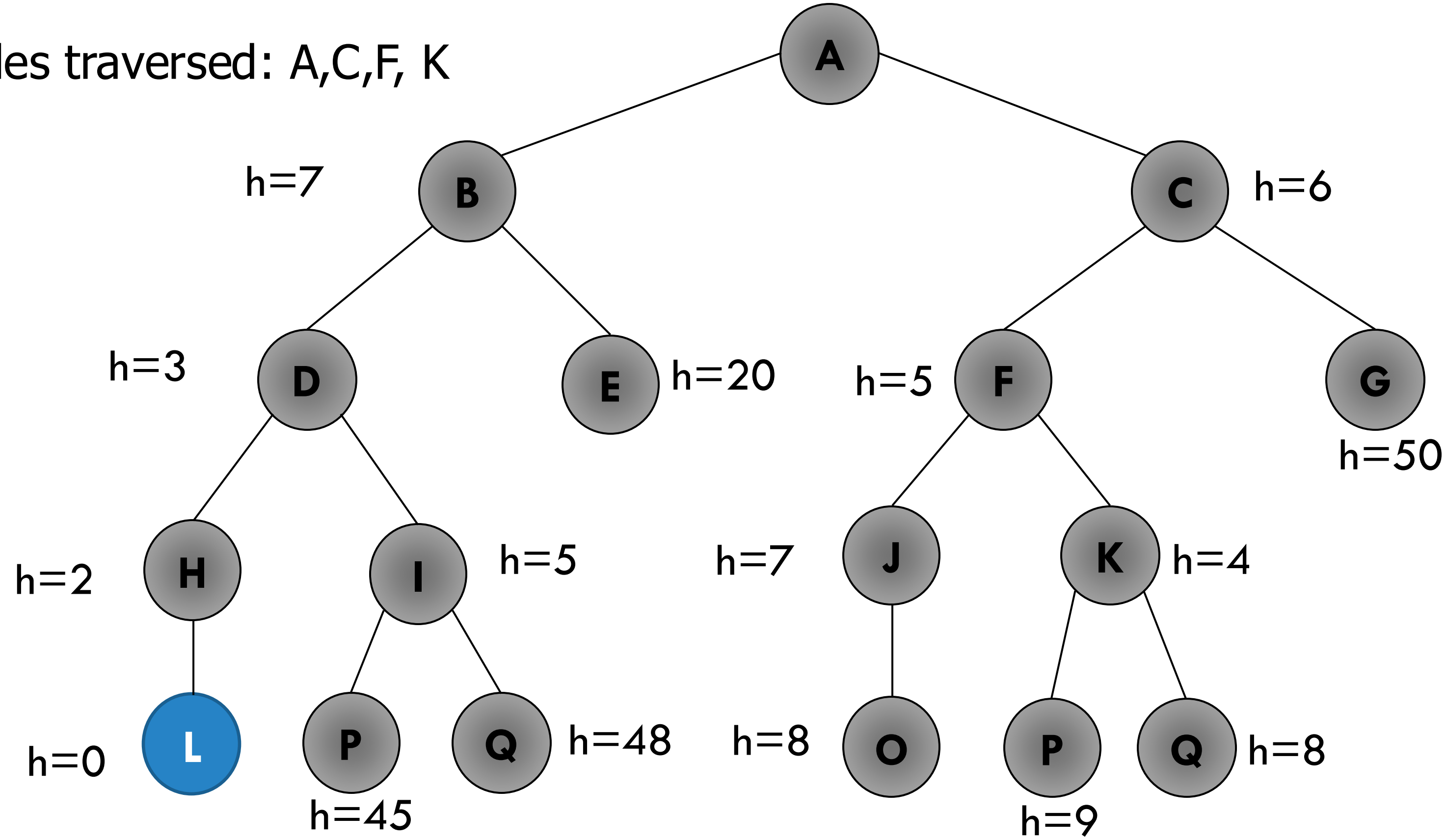


HILL CLIMBING SEARCH

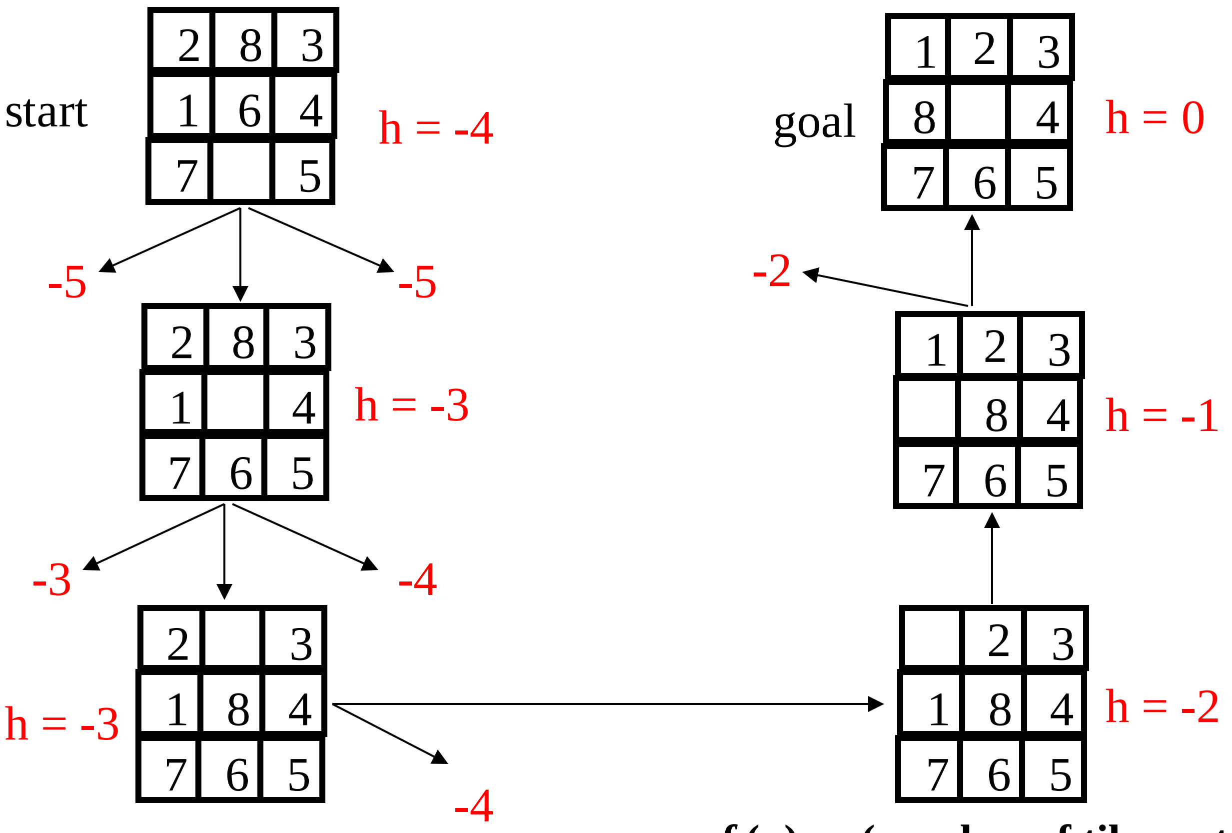
1. Hill climbing can be applied to the travelling salesman problem.
2. It is easy to find an initial solution that visits all the cities but will be very poor compared to the optimal solution.
3. The algorithm starts with such a solution and makes small improvements to it, such as switching the order in which two cities are visited.
4. Eventually, a much shorter route is likely to be obtained

HILL CLIMBING SEARCH

Nodes traversed: A,C,F, K



HILL CLIMBING EXAMPLE



f **(*n*) = -(number of tiles out of place)**

TYPES OF HEURISTICS

Perfect heuristic

If $h(n) = h^*(n)$ for all n, then only the nodes on the optimal solution path will be expanded. So, no extra work will be performed.

Null heuristic

If $h(n) = 0$ for all n, then this is an admissible heuristic and A* acts like Uniform-Cost Search.

Better heuristic

If $h1(n) < h2(n) <= h^*(n)$ for all non-goal nodes, then h2 is a better heuristic than h1

- If A1* uses h1, and A2* uses h2, then every node expanded by A2* is also expanded by A1*.
- In other words, A1 expands at least as many nodes as A2*.
- We say that A2* is better informed than A1*.

<u>The closer h is to h*, the fewer extra nodes that will be expanded</u>

TYPES OF HEURISTICS

5		8
4	2	1
7	3	6

n

1	2	3
4	5	6
7	8	

goal

h1(n) = hamming distance: number of misplaced tiles = 6 is admissible

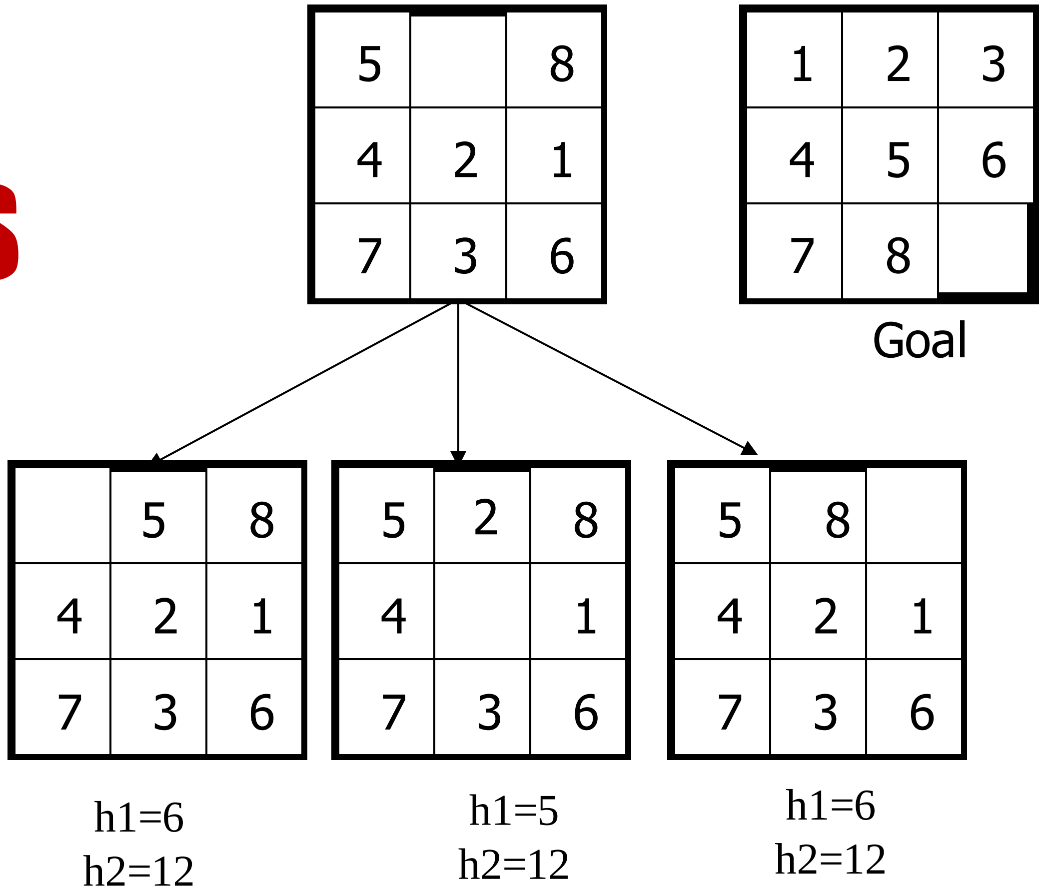
h2(n) = Manhattan Distance: sum of distances of each tile to goal = 13 is admissible

Let h1 and h2 be two admissible and consistent heuristics such that for all nodes N: h1(n) $\leq$ h2(n).

Then, every node expanded by A* using h2 is also expanded by A* using h1. h2 is more informed than h1

h1(n) < h2(n) both are admissible, h2 is more informed than h1

BREAKING TIES



HOW TO OBTAIN ADMISSIBLE HEURISTICS?

Can be derived by relaxing the problem

If the rules of the 8-puzzle are relaxed so a tile can move anywhere, then h1 (hamming distance) gives the shortest cost to the goal.

If the rules are relaxed so the tile can move to any adjacent square, then h2 (Manhattan distance) gives shortest cost.

POSSIBLE A* HEURISTICS

3D Manhattan distance

Max Manhattan distance (not admissible): computes the maximum between the sum of the Manhattan distances of all corners from their right position and orientation, and the sum of the Manhattan distances of all edges from their right position and orientation.

Sum Manhattan distance : Computes the Manhattan distance of every tile to to its original place in the face it belongs to (the face with the same color in the middle tile) and sums the results. in order to make it admissible, we divided the result by 8 since every twist moves 8 cubies.

POSSIBLE A* HEURISTICS

Colors union: this heuristic creates an abstraction of the original state. we tried four different abstractions: $f\ or\ n \in \{2, 3, 4, 5\}$ the abstraction merges every combination of n colors into one color. For example, 2 colors abstraction creates (62) = 15 cubes with 5 different colors (paints one face with one of the other faces' color) Then we run BFS on every abstracted state and return the maximum of the solution lengths found by every BFS run.

Number of misplaced tiles: sums the number of tiles that are not located in their place, for every face, and normalize the result by 6 (the number of faces). Because the minimal changes that can occur in the number of misplaced tiles is 3, dividing by 6 keeps the

Ignoring corners: this heuristic creates an abstraction of the original state. for each tile in each corner, replace it with the color of the tile in the center of the face (the 'target' color)

GOD's Algorithm

God Algorithm

Any algorithm which produces a solution with the fewest possible moves. Only an Omniscient Being knows the optimal step moving forward in the process.

That is, starting from any initial configuration, the algorithm is able to determine a sequence of optimal actions leading to reach a target solution (*if* the problem is solvable from that initial configuration.

A solution is optimal if the sequence of moves is as short as possible.

The count of moves is known as **God's number**, or, more formally, the **minimax** value.

God's algorithm: for a given puzzle, is an algorithm that solves the puzzle and produces only optimal solutions.

UNSOLVED GAMES

Some well known games have a limited set of simple well-defined rules and moves, but have nevertheless never had their God's algorithm for a winning strategy determined (e.g., chess and Go).

The total number of all possible positions, approximately 10^{154} for chess
and 10^{180} (on a 19×19 board) for Go.
.
too large for a brute force solution with current computing technology,

In contrast: compared, the now solved, with great difficulty, Rubik's Cube which involves only 4.3×10^{19} configurations.

God's algorithm and difficult games

While chess computers have been built that are capable of beating even the best human players, they do not calculate the game all the way to the end.

Deep Blue, for instance, searched only 11 moves ahead (counting a move by each player as two moves), reducing the search space to only **10^{17}**.

Further, each position for advantage according to rules derived from human play and experience.

Why not evolutionary?

Evaluation algorithms are prone to make elementary mistakes so even for a limited look ahead with the goal limited to finding the strongest interim position, a God's algorithm has not been possible for Go.

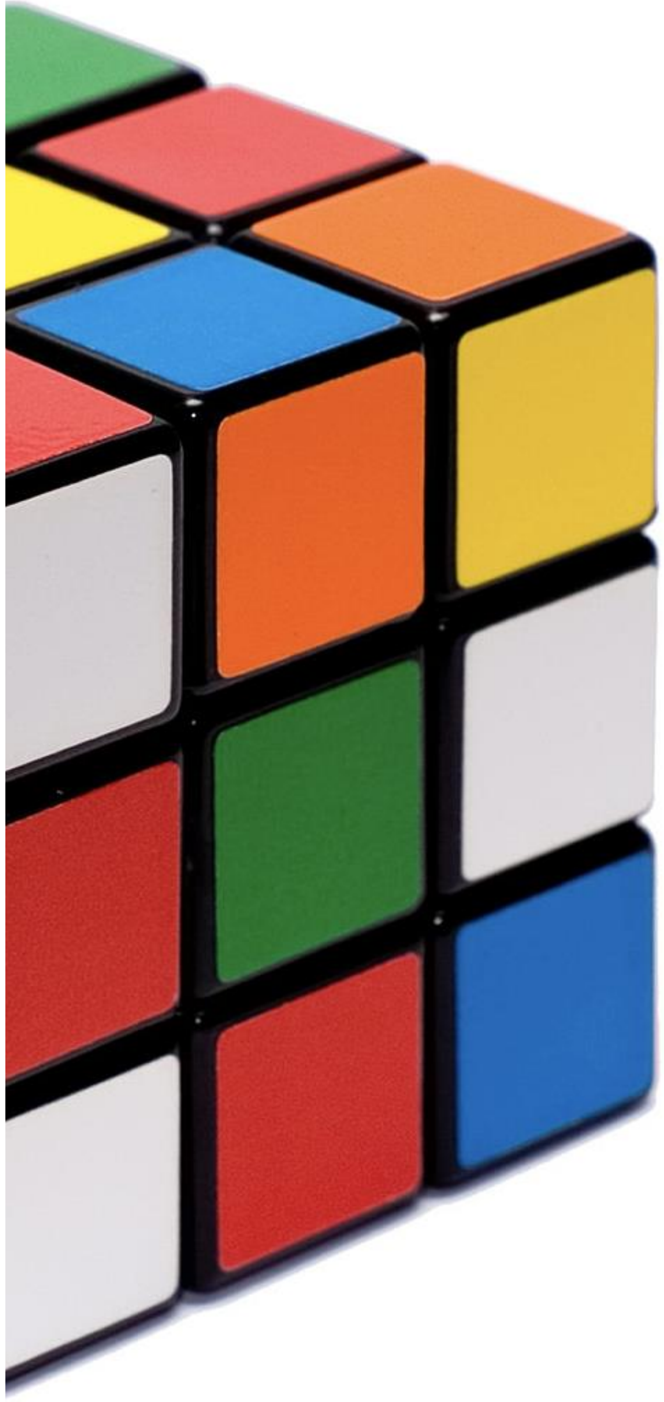
Rubik's cube and god's algorithm

Rubik's Cube 1

Rubik's Cube invented in the late 1970s by Erno Rubik, is another known challenge for a single-agent search.

Rules 2

Each face can be rotated by 90, 180, or 270 degrees and the goal is to rearrange the subcubes, called *cubies*, of a scrambled cube such that all faces are uniformly colored.



NUMBER OF CONFIGURATION

Description

There are 6 centers, each of a distinct color.

Then there are 12 edges and 8 corners, which are movable and hence, enabling the permutations of the Rubik's Cube

So, the number of ways to arrange these 8 corners is 8! *i.e.* 40,320.

Each corner has actually 3 different possible configurations (White-Red-Green and Red-Green-White being the other two configurations for our corner).

we can only orient 7 corners independently. That is the orientation of the eighth corner will get fixed automatically depending on the orientations of the remaining seven corners.

Hence, the number of permutations arising from the 8 corners is- $8! \times 3^7$.

The number of ways to arrange these 12 edges is 12! *i.e.* 479001600.

Each edge is made of two different colors and hence, can have two different configurations.

we can only orient 11 of the 12 edges independently.

The twelfth edge will get oriented automatically. Hence, the number of permutations arising from the 12 edges is- $12! \times 2^{11}$.

In rearranging the 8 corners or the 12 edges, we need to take into account an important thing and that is we cannot swap two corners or two edges in isolation without affecting the neighboring pieces.
We will never have a cube in a solved state with except only two of its edges or corners swapped.

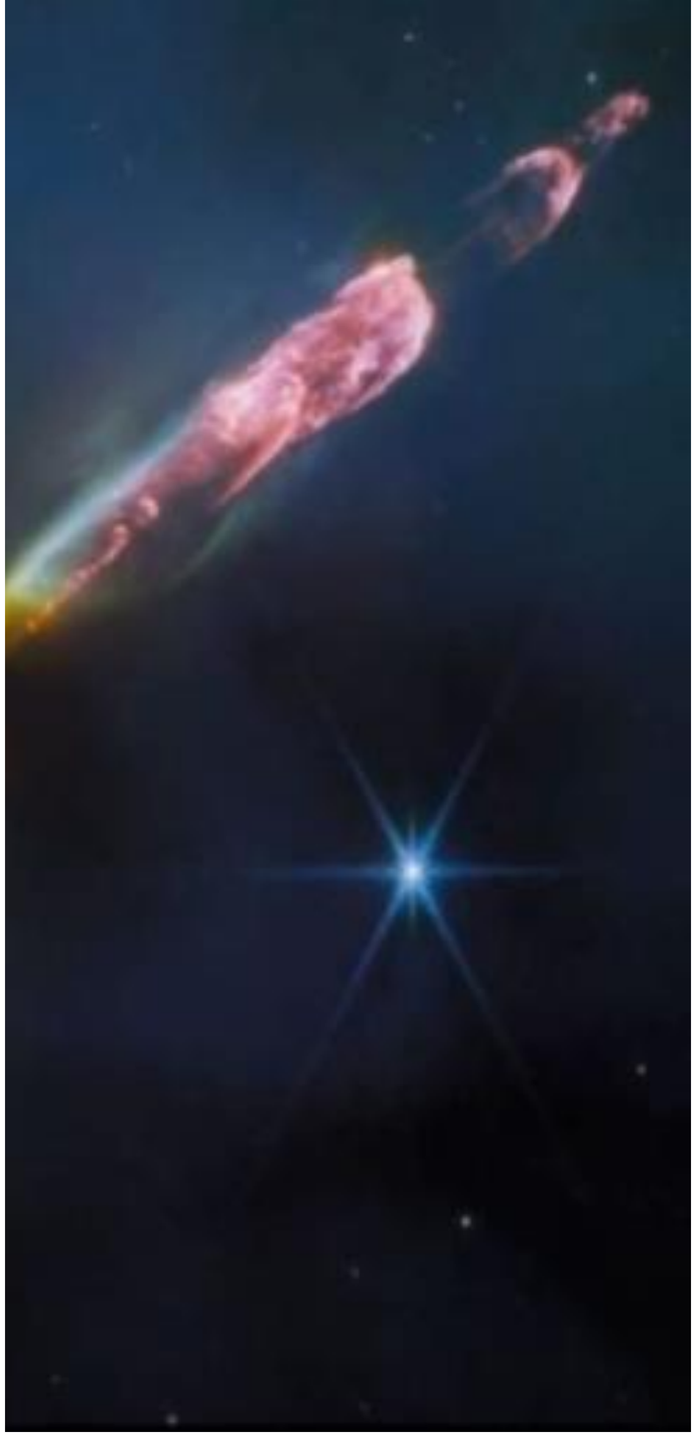
But we have actually counted these impossible states as well. So, we will actually have only half of the permutations we have calculated.

the total number of possible permutations of the Rubik's cube is:

$(1/2) * (8! \times 3^7) * (12! \times 2^{11})$
=
43,252,003,274,489,856,000.

Approx 4.3×10^{19}

The SEARCH SPACE

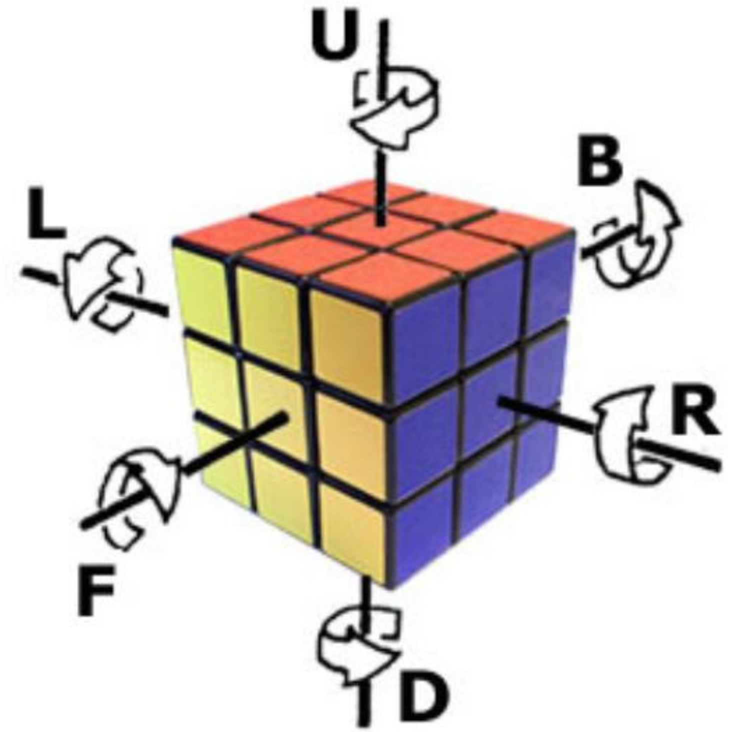


REPRESENTING THE MOVES

Three different ways to represent the moves, which can be further divided into two sub moves for each movement.

- a move is considered to be a 90, 180 or 270 degree rotation relative to the rest of the cube.

- A single letter L : Rotate Left face 90 degrees clockwise.
- A Letter followed by apostrophe: e.g, L': Rotate the left face 90 degrees counter clockwise.
- A letter with the number 2 after: e.g, L2: Rotate the left face 180 degrees.



For example LU'D2 can describe the following steps:

1.Rotate left face 90 degrees clockwise.
2.Rotate upper face(top face) counterclockwise
3. Rotate the Down(bottom) face

Pattern Database

- Heuristics: Pattern Databases (PDBs) offer good heuristics for Rubik's Cube.
- For instance, if we ignore the edges and exhaustively solve for all corners, and store the results in a hash table.
- A perfect hash function can result in unique compact mapping which is very memory efficient.
- There are 88 million combinations and less than 16 values, you can store this in 44 MB of memory.
- Retrieving a heuristic for a state, we use the hash function to look up the corner configuration in the table, which contains the total number of moves required to solve that configuration. That is an admissible (and consistent) heuristic for the problem.

- One may consider doing the edges, but the 12-edge PDB takes 500GB of memory to store and might not fit in memory. Hence, we may do subsets of edges.
- We may also consider using the cube symmetries and many other patterns to get better heuristic values.
- With a good parallel IDA* implementation and some large PDBs you can solve random Rubik's cube instances optimally.

Comparison

IDA* for each heuristic, and compared between the time taken to find a solution, and number of nodes expanded.

Colors union heuristics were the slowest heuristics. union of 2,3 and 4 colors took over 4 minutes and 5 colors union over a minute, only on 2 actions and for more actions they didn’t stop.

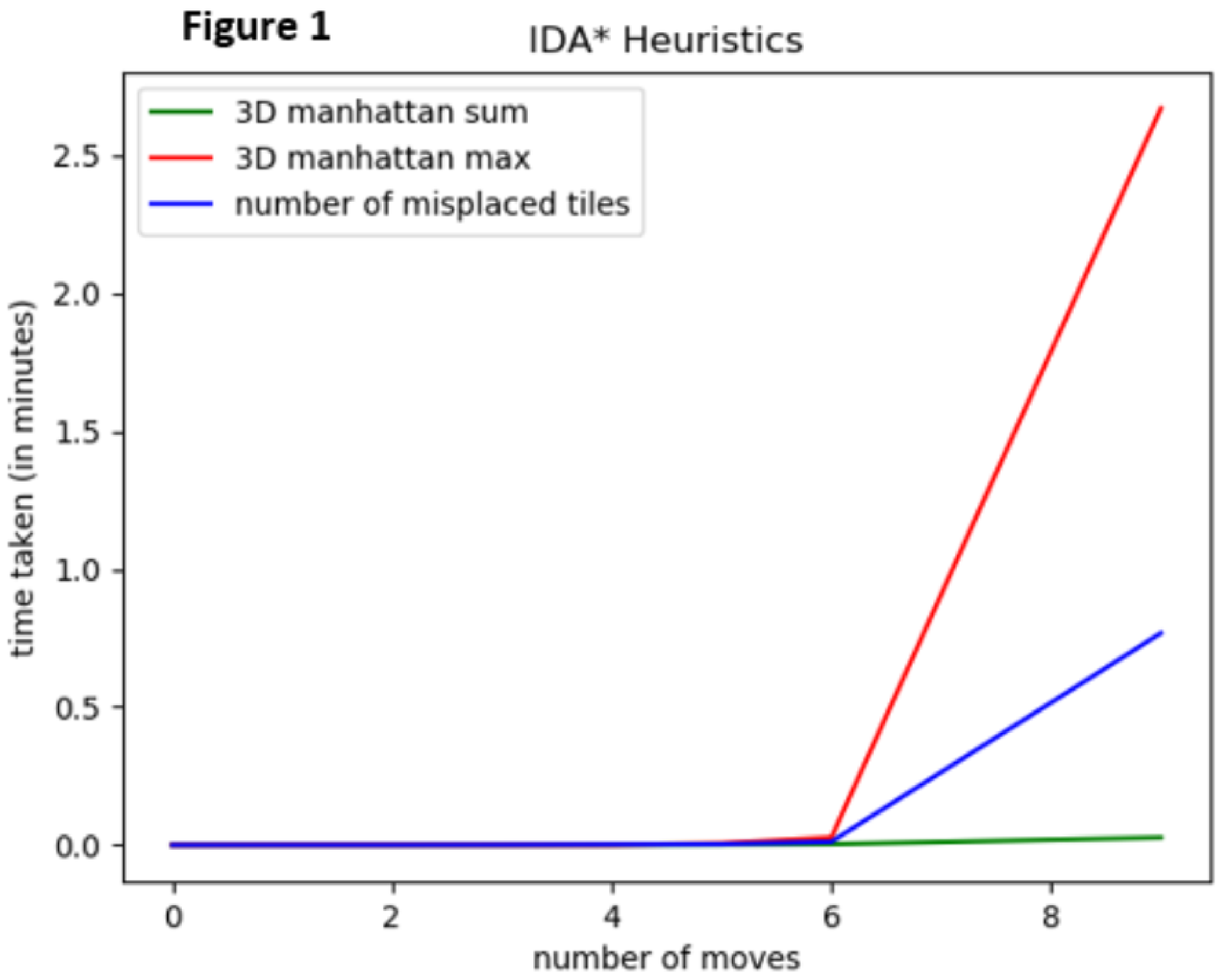
Max manhattan heuristic, number of misplaced tiles heuristic, and sum manhattan heuristic were the best heuristics.

Max manhattan heuristic could solve cube with no more than 9 steps shuffling,

Number of misplaced tiles heuristic could solve cube with no more than 10 steps

Shuffling and sum manhattan heuristic could solve cube with no more than 12 steps shuffling.

Computational Time Performance



Optimal Solutions for Rubik's Cube

An algorithm to determine the minimum number of moves to solve Rubik's Cube was published in 1997 by Richard Korf.

While it had been known since 1995 that 20 was a lower bound on the number of moves for the solution in the worst case, it was proven in 2010 through extensive computer calculations that no configuration requires more than 20 moves.

Thus 20 is a sharp upper bound on the length of optimal solutions. Mathematician David Singmaster had "rashly conjectured" this number to be 20 in 1980.

A HISTORY OF GOD'S NUMBER

With about 35 CPU-years of idle computer time donated by Google, a team of researchers has essentially solved every position of the Rubik's Cube™, and shown that no position requires more than twenty moves.
Each twist of any face is considered to be one move (this is known as the half-turn metric.)
Every solver of the Cube uses an algorithm, which is a sequence of steps for solving the Cube. One algorithm might use a sequence of moves to solve the top face, then another sequence of moves to position the middle edges, and so on.
There are many different algorithms, varying in complexity and number of moves required, but those that can be memorized by a mortal typically require more than forty moves.
The number of moves this algorithm would take in the worst case is called God's Number.
At long last, God's Number has been shown to be 20.

Fifteen years after the introduction of the Cube to find the first position that provably requires twenty moves to solve; it is appropriate that fifteen years after that, we prove that twenty moves suffice for all positions.

Date	Lower bound	Upper bound	Gap	Notes and Links
July, 1981	18	52	34	Morwen Thistlethwaite proves 52 moves suffice.
December, 1990	18	42	24	Hans Kloosterman improves this to 42 moves.
May, 1992	18	39	21	Michael Reid shows 39 moves is always sufficient.
May, 1992	18	37	19	Dik Winter lowers this to 37 moves just one day later!
January, 1995	18	29	11	Michael Reid cuts the upper bound to 29 moves by analyzing Kociemba's two-phase algorithm.
January, 1995	20	29	9	Michael Reid proves that the "superflip" position (corners correct, edges placed but flipped) requires 20 moves.
December, 2005	20	28	8	Silviu Radu shows that 28 moves is always enough.
April, 2006	20	27	7	Silviu Radu improves his bound to 27 moves.
May, 2007	20	26	6	Dan Kunkle and Gene Cooperman prove 26 moves suffice.
March, 2008	20	25	5	Tomas Rokicki cuts the upper bound to 25 moves.
April, 2008	20	23	3	Tomas Rokicki and John Welborn reduce it to only 23 moves.
August, 2008	20	22	2	Tomas Rokicki and John Welborn continue down to 22 moves.
July, 2010	20	20	0	Tomas Rokicki, Herbert Kociemba, Morley Davidson, and John Dethridge prove that God's Number for the Cube is exactly 20.