Rubik's Cube Solving Algorithms

Srushti HP-181IT118 (8105813574)
Seema G-181IT241 (9483852115)
Shonali KS-181IT244 (6362295510)
Ravi Prakash- 181IT137 (7667123460)

Introduction

Algorithms Used:

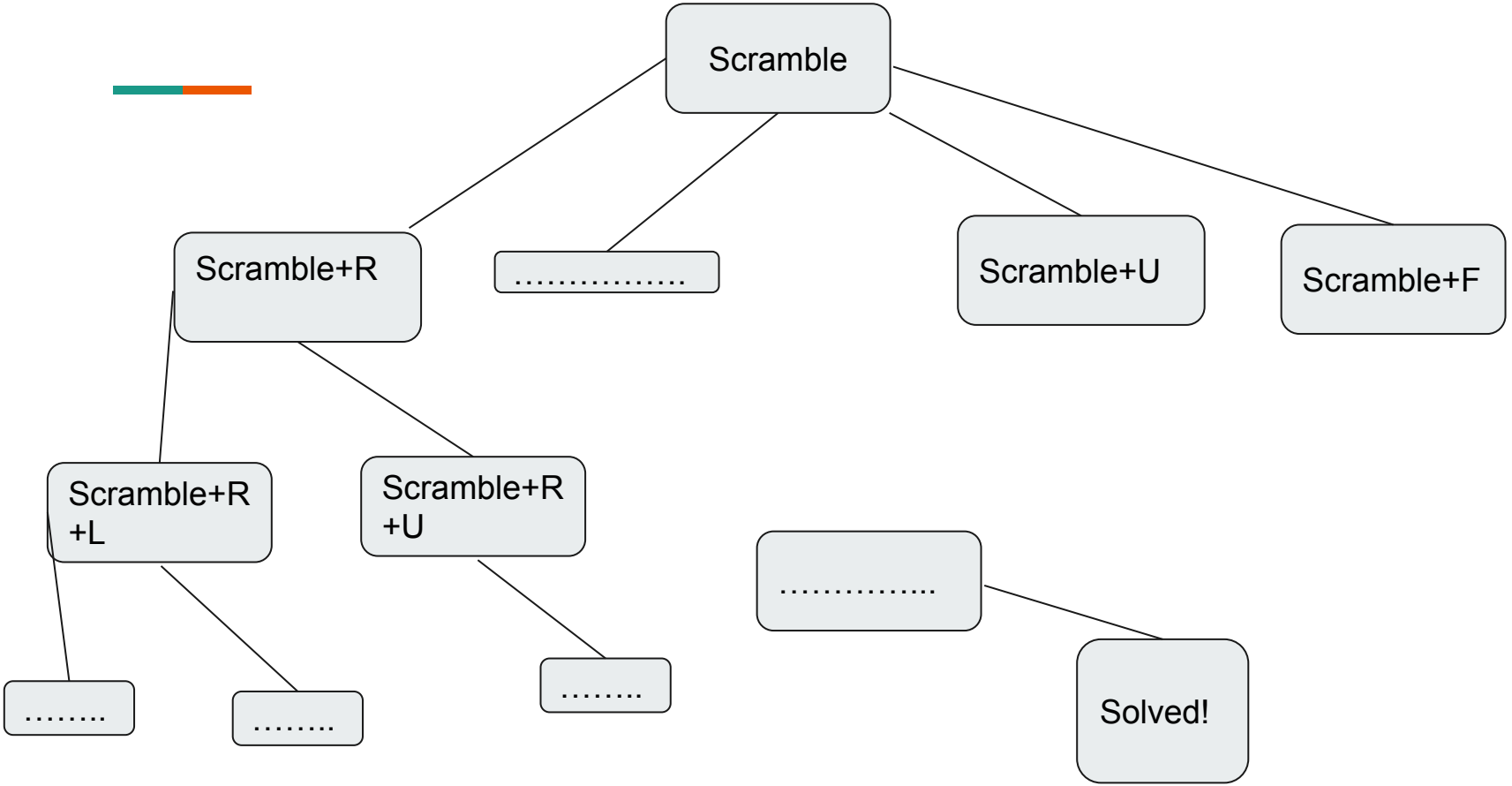
1. IDFS- Iterative Depth First Search
2. IDA*- Iterative Deepening A* also known as Korf's Algorithm
    a. Two Heuristics are used for IDA*
        i. IDA* with maximum of the sum of manhattan distance of cube corners and of edges as heuristics
        ii. IDA* which uses depth of corners configuration of cube as heuristic

Important Terms

Some terms we have used are:

Goal Height: Depth of the solution from root node (i.e Number of moves)

Branching Factor: No of children at each node i.e the outdegree


Scramble
Scramble+R
…….
Scramble+U
Scramble+F
Scramble+R
+L
Scramble+R
+U
……..
……..
……..
……..
Solved!

Iterative Depth First Search (IDFS)

- **IDFS** combines depth-first search's space-efficiency and breadth-first search's fast search (for nodes closer to root).
- If 'd' is depth and 'b' is the branching factor of search tree,Runtime of IDFS is assymptotically $O(b^d)$.
- But the constants are large and hence slower than BFS and DFS.

Iterative Deepening A* (IDA*)-Heuristics 1(h1)

- It takes the maximum of the sum of manhattan distance of cube corners and sum of manhattan distance of edges.

- 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.

Iterative Deepening A* (IDA*)-Heuristics 2(h2)

- It uses depth of corners configuration of cube as heuristic. Depth is saved in **Pattern Database.**
- Convert the state to the abstracted state according to the chosen abstraction -search for the abstracted state in the DB-return the value saved in this location in the DB
- BFS implementation is not efficient enough,and building the db to depth 8 can take almost a week, hence our db is restricted to depth 4.
- It’s a permissible heuristic because it never overestimates the distance to the solved state.

Results

Depth		Nodes generated			Branching Factor			Time taken (s)	
	IDFS	IDA*-h1	IDA*-h2	IDFS	IDA*-h1	IDA*-h2	IDFS	IDA*-h1	IDA*-h2
1	12	24	12	12	1	1	0.0009	0.0114	0.0507
2	84	24	24	11.16	1	1	0.0065	0.0115	0.0458
3	528	36	36	10.8	1.333	1.333	0.0543	0.0193	0.0417
4	18036	240	48	10.86	2.25	1.25	2.20	0.1187	0.0455
5	86136	3828	Computationally intense	10.854	3.3020	Computationally intense	13.200	1.666	Computationally intense

Conclusion

- IDFS and IDA* algorithms were able to solve the cube.
- IDFS time increased rapidly with depth of solution. IDA* performed quite faster as compared to IDFS.
- IDA* worked even faster with pattern database.

Refernce :https://www.cs.huji.ac.il/~ai/projects/2017/heuristic_%20search/rubiq/files/ai_report.pdf