UNIVERSITY OF NEW HAVEN

Solving the Rubik's Cube

Reinforcement learning for the Rubik Cube

A Team Project By:-

BHANU PRATAP (00760641)
KRISHNANJALI DAMERA(00808762)
PRASHANT RANA (00804232)

Presentation plan

- Project objectives
- Our approach
- Deliverables
- Evaluation
- Conclusion

Statement of Project Objective

Project Objective:

- The primary goal of the project is to solve the Rubik's cube issue. This is accomplished by using a pattern database and effective reinforcement technique, to measure the quality of nearly finished cubes.
- When compared to the more popular 3x3x3 cube, the 2x2x2 cube is much easier to solve.
- This project likewise aims to solve the problem of the cube with n random motions applied to it and increases its efficiency in solving the given Rubik's cube.

Our Approach

- Bivariate Breadth-First Search (BFS) is a graph traversal algorithm that explores a graph from two starting points simultaneously, one moving forward and the other moving backward, with the goal of finding the shortest path between these two points. It is often used in scenarios where you need to find the shortest path between two nodes or vertices in an unweighted or weighted graph.

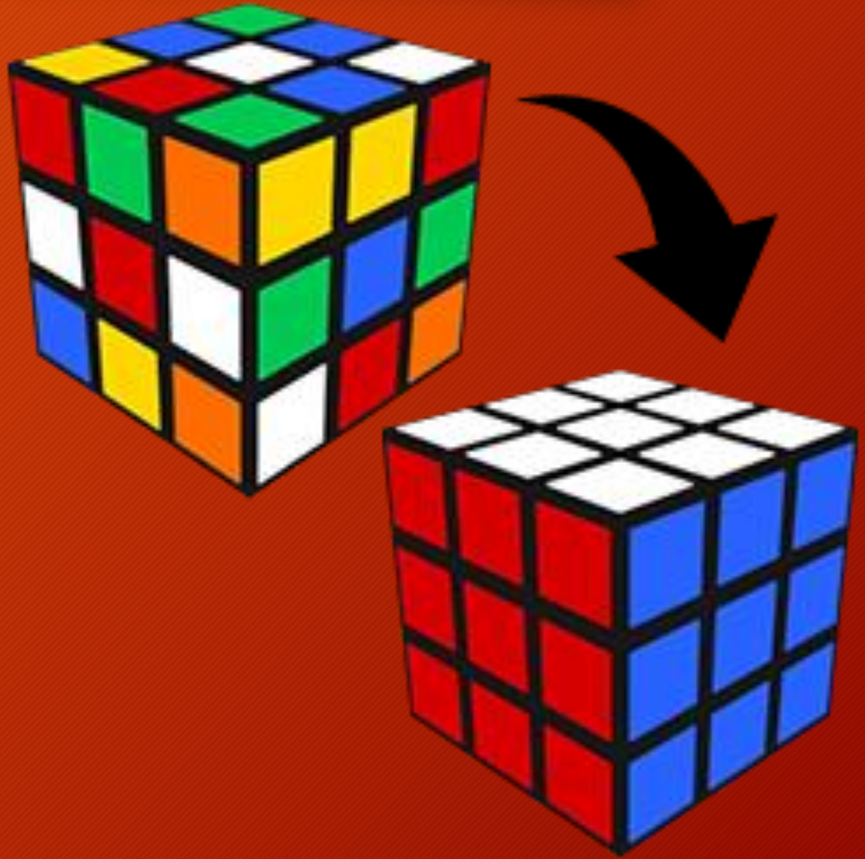
- Tools and Techniques used are:
  - Python

Deliverables

- A Project Proposal
- Project code
- A video Demonstration

Evaluation

- The outcome should be checked to judge the ability of the project to Solve a simple Rubik's cube that consists of 2x2x2 or 3x3x3 cubes in an efficient manner.
- Evaluating the algorithm's performance involves considering its completeness, efficiency, memory usage, and scalability, as well as any optimizations or heuristics used to enhance its performance.



Implementation

- The Rubik's Cube is represented as a tuple of 24 integers, where each integer represents the color of a sticker on a specific face of the cube. The order of stickers corresponds to the solved state. The algorithm maintains separate queues, distances, and parent dictionaries for forward and backward searches.

Algorithm

- Breadth first search

BFS is a tree traversal algorithm, in which the nodes at current level are checked before proceeding to next level. Our shortest_path(start, end) function implements BFS. In this function, we provide the start state and the end state for solving our Rubik's Cube.

- Bidirectional breadth-first search

Bidirectional BFS is a graph search algorithm, which finds the smallest path by traversing from start and end state using BFS. Our shortest_path_optimized(start, end) function implemented bidirectional BFS. Here also, we provide the start state and the end state for solving our Rubik's Cube.

Optimization details

- Bidirectional BFS often reduces the time complexity by exploring fewer states compared to a unidirectional BFS.

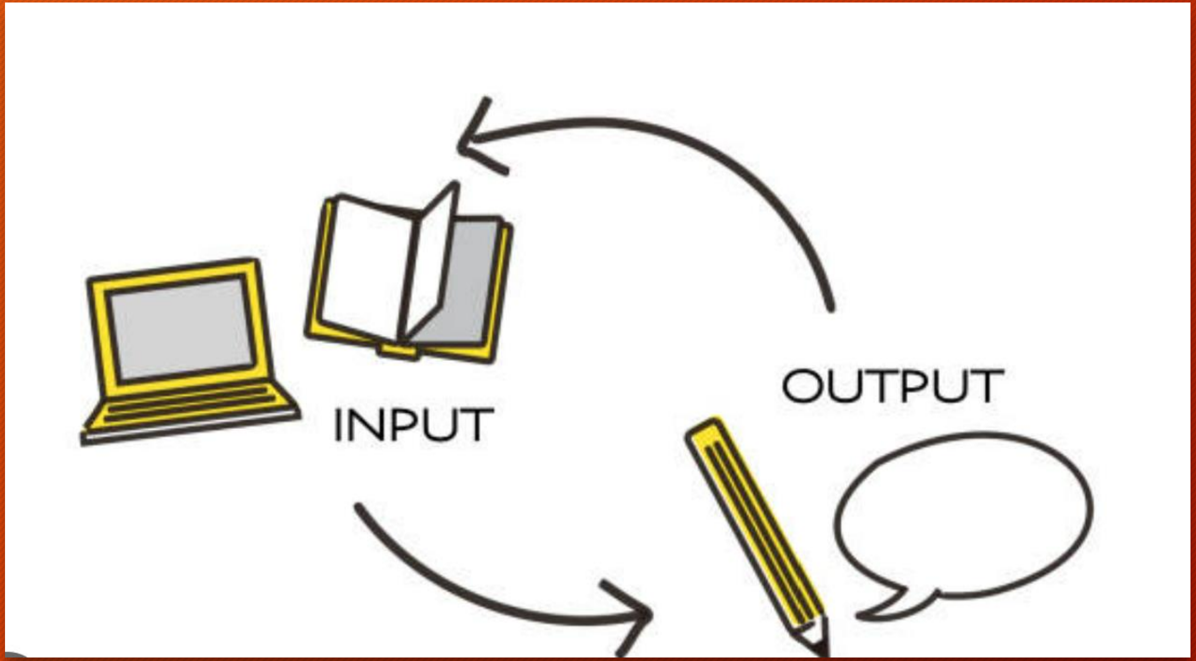
- By searching from both ends, the algorithm narrows down the search space more quickly.

Path Reconstruction

- The found path is reconstructed by tracing back from the end state to the start state using the visited or start_parent and end_parent dictionaries.
- Highlight the bidirectional exploration of states from both the start and end, aiming for convergence in the middle.

Output

- Our project aims to find the shortest path and increase the efficiency of finding the shortest path. Hence, our project returns the shortest path found using BFS and also calculates the time taken to find the shortest path. Also, we use bidirectional BFS to find the shortest path which may or may not be the same path found by BFS, and the time taken to find the shortest path.



Conclusion

- Bivariate BFS optimizes the search process by leveraging a dual-front approach, reducing the search space, and providing a balance between time and space complexity. It is particularly effective in scenarios where the goal is to find the shortest path, as it ensures both efficiency and optimality in pathfinding
- Bivariate BFS for the Rubik's Cube is a significant advancement that showcases the power of AI in solving complex, real-world puzzles. It holds promise for a wide range of applications beyond the Rubik's Cube, contributing to the broader field of reinforcement learning and artificial intelligence.

Thank You