AI-Based Rubik's Cube Solver

Anthony Santos: anthony.santos@unh.edu

CS 730 / 830
Introduction to
Artificial Intelligence

University of New Hampshire

Introduction

- ❑ The Rubik's Cube is a classic problem involving planning, searching, and learning.
- ❑ Humans use structured methods and strategies like CFOP.
- ❑ Can AI techniques replicate or outperform human efficiency?
- ❑ The goal is to implement, train, and compare, multiple solving strategies across fairness-controlled scrambles using shared metrics.

Data Set

- ❑ Input: Scrambled 3x3x3 Rubik's Cube (randomized 3 moves)
- ❑ Output: Move sequence to solve
- ❑ Scrambles: Random 3-move sequences generated uniformly
- ❑ State Representation: Cube: 54-element vector (6 faces x 9 stickers)
- ❑ Actions: 18 discrete face rotations
- ❑ All solvers tested on identical scrambles for fairness

Code & Data

- ❑ https://github.com/ajs1288/rubiks_ai

Methods

- ❑ Human Method:
  - ❑ CFOP (via PyCuber): Speedcubing method: Cross→F2L→OLL→PLL
- ❑ Classical AI Solvers:
  - ❑ BFS: Exhaustive search, optimal for ≤3 moves
  - ❑ A* Search: Uses heuristic (misplaced facelets) to guide search
- ❑ Reinforcement Learning (DQN):
  - ❑ Agent: Trained using a custom Gym-style environment with reward shaping
  - ❑ State: Flattened 54-element vector of sticker colors
  - ❑ Action Space: 18 face moves
  - ❑ Reward:
    - ❑ +100 for solving
    - ❑ Otherwise: reward=1-(distance/54)

Future Work

- ❑ Train RL agent 5-10 move scrambles with curriculum learning over a longer period-of-time.
- ❑ Integrate 2D viewer to replay solutions.

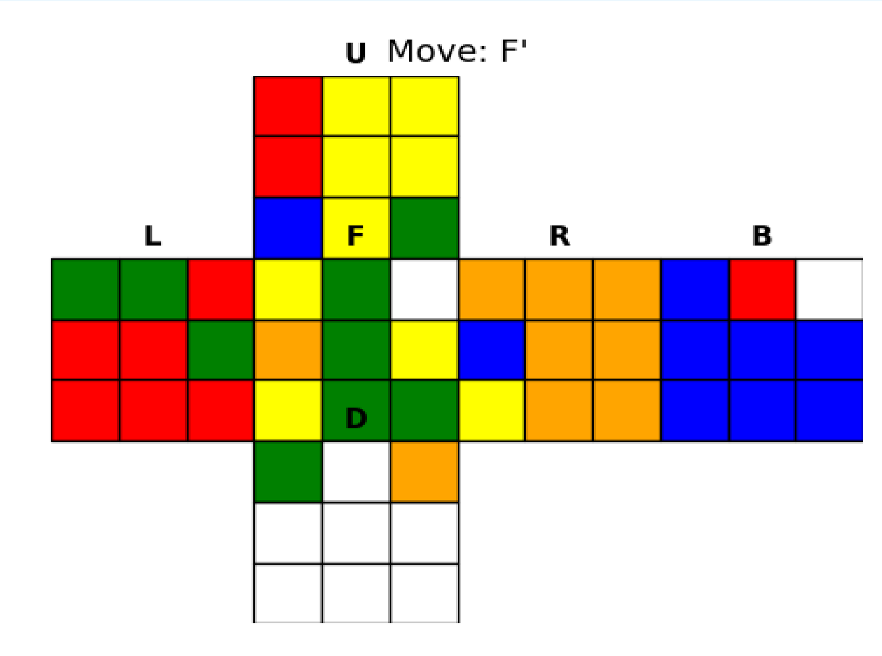

Figure 1: In-progress 2D Viewer

Evaluation Results

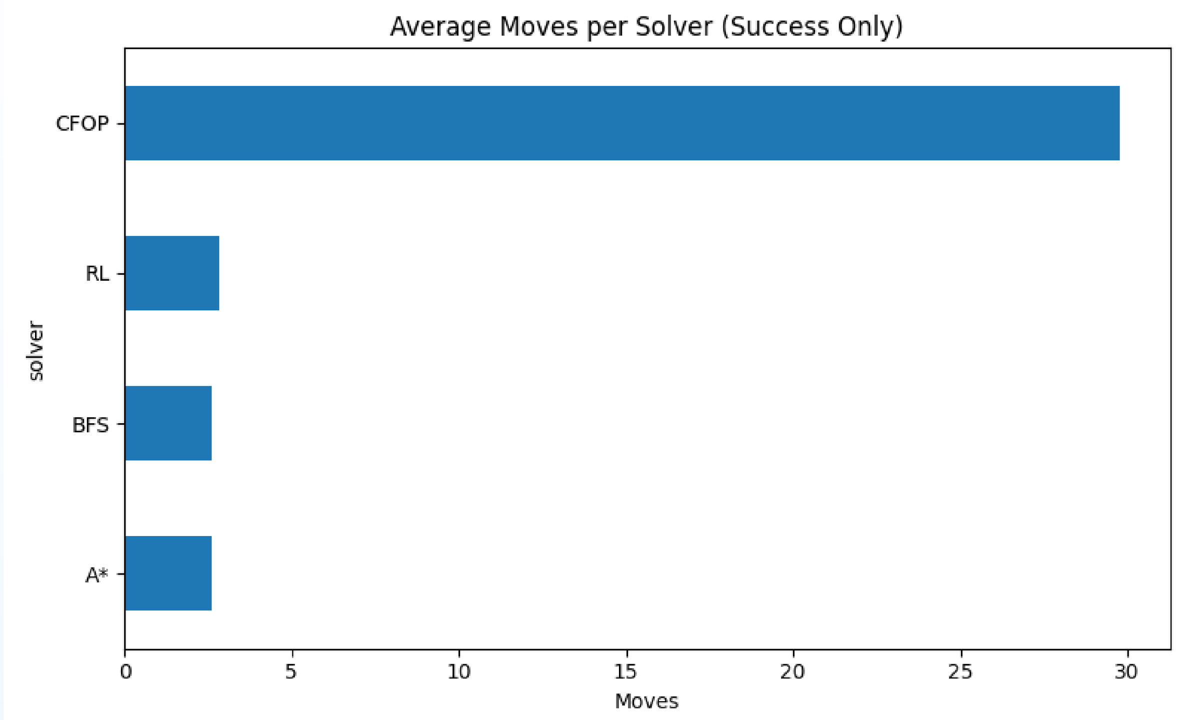

Figure 2: Average number of moves per solver

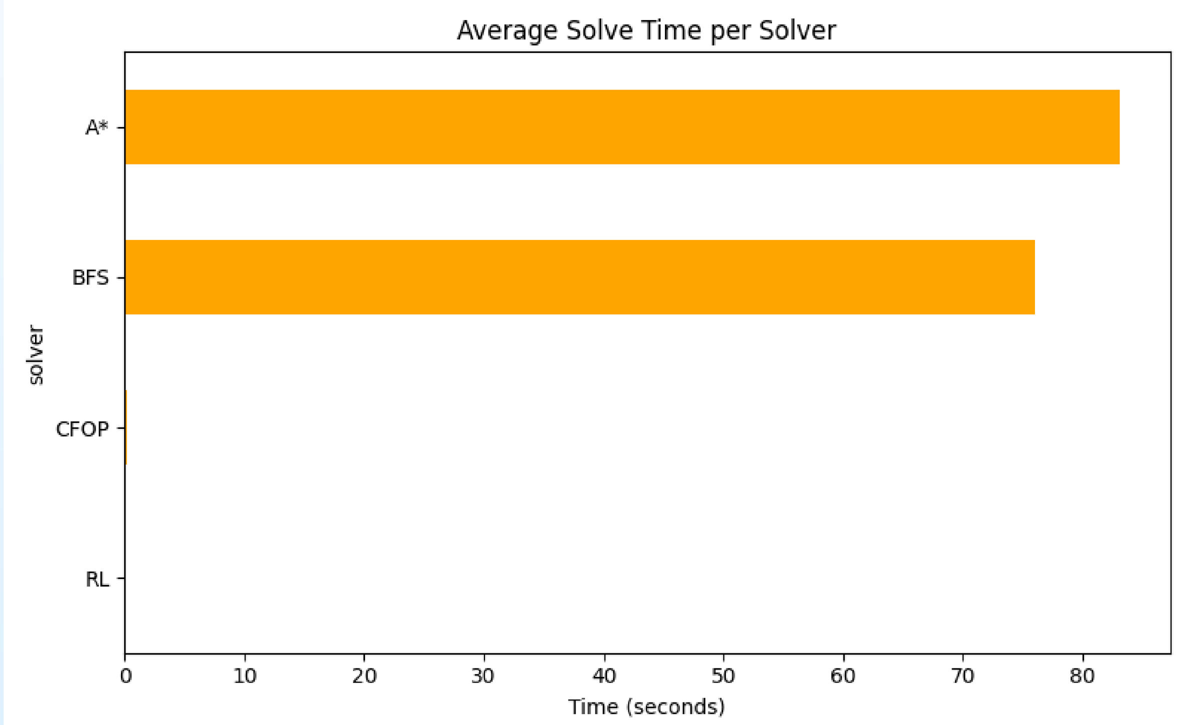

Figure 3: Average solve time per solver

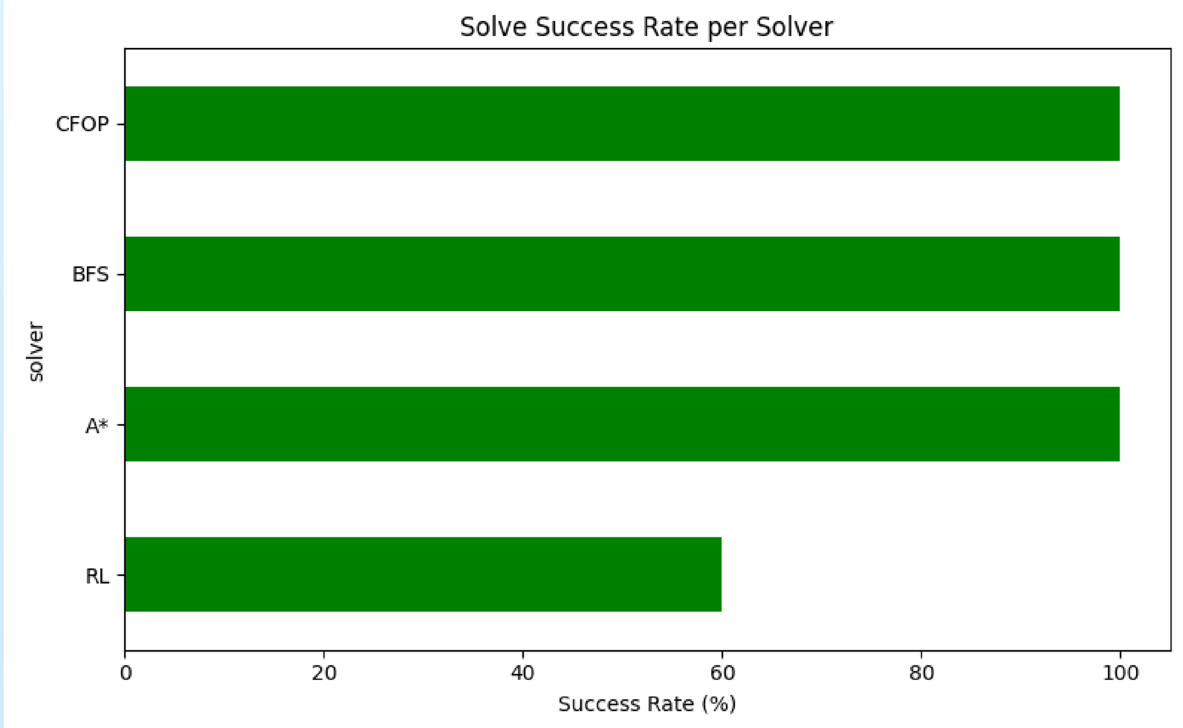

Figure 4: Solve success rate

- ❑ CFOP and RL are much faster than BFS and A*
- ❑ BFS, A*, and RL used fewer moves than CFOP
- ❑ RL occasionally failed
- ❑ The AI solvers were efficient but unreliable.
- ❑ CFOP isn't optimal but is consistent.