Solving the Rubik's Cube with a PDDL Planner

Bharath Muppasani, Vishal Pallagani, Biplav Srivastava

AI Institute, University of South Carolina

Demo Video

Introduction

- The Rubik's Cube (RC), invented in 1974 and popular since its inception, is a 3D puzzle game consisting of 26 colored pieces anchored to a central spindle, with the goal of manipulating it to display a uniform color on each face.
- The puzzle incorporates primary rotations - Up (U), Down (D), Right (R), Left (L), Front (F), and Back (B) - each representing a 90-degree clockwise turn, while their inverses indicate a counter-clockwise rotation.
- The Rubik's Cube is a prime example of a combinatorial puzzle with over 43 quintillion possible states. Navigating this immense state space efficiently poses significant computational challenges, especially for time and memory constraints.
- Using PDDL models allows leveraging powerful off-the-shelf planners to solve complex puzzles, saving time and resources by avoiding the need to develop custom solvers.
- PDDL provides a standardized problem representation, enhancing scalability and flexibility, making it easier to apply to various combinatorial puzzles and real-world applications

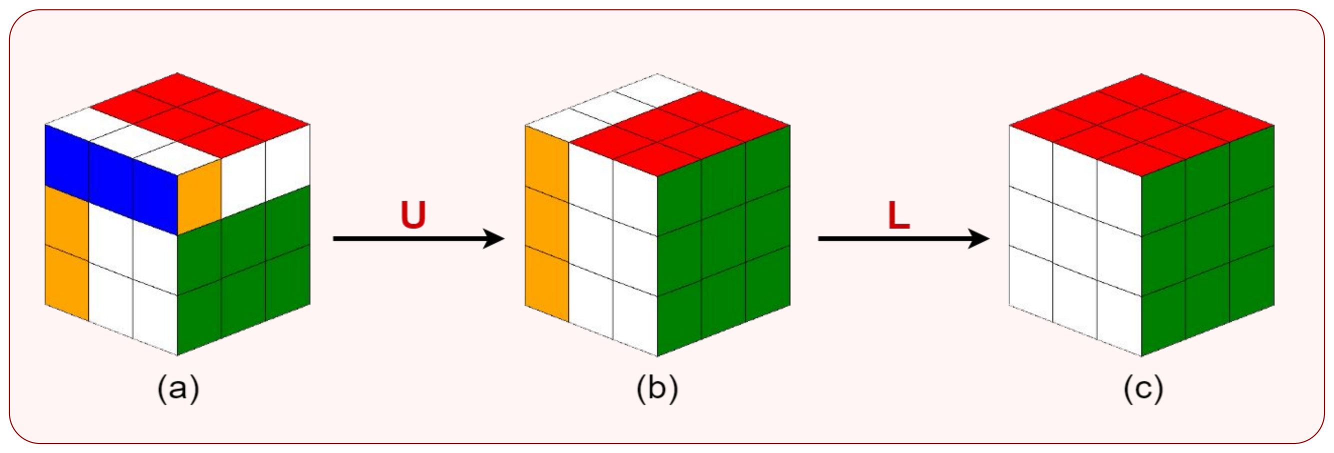

Figure-1: Shuffled state of the cube. Solution found with FD planner: U, L. Cost: 2.

Future Work

- Future studies should explore into various other domains, comparing planning representations to achieve a comprehensive insight.
- Highlight the potential of integrating learning mechanisms within domain-independent methods for improved problem-solving.
- Encourage collaborations between the planning and learning communities to devise novel methodologies that combine the strengths of both fields.

References

- **Demo**: Muppasani, B., Pallagani, V., and Srivastava, B., 2024. Solving the Rubik's Cube with a PDDL Planner. Accepted at the 24th International Conference on Automated Planning and Scheduling (ICAPS 24) Demo Track.
- **Comparative Anlysis**: Muppasani, B., Pallagani, V., Srivastava, B. and Agostinelli, F., 2024. On Solving the Rubik's Cube with Domain-Independent Planners Using Standard Representations. arXiv preprint arXiv:2307.13552. *(Accepted at HSDIP workshop ICAPS-24)*

Rubik's Cube PDDL

In the PDDL domain, the Rubik's cube environment is defined with fixed cube pieces named as shown in Figure 2. These pieces are modeled as predicates in the RC domain, with their colors in three-dimensional space as predicate parameters. We use conditional effects to define each action as a change in colors on these fixed cube pieces.

```
(:action L
:effect (and
;for corner cubelets
(forall(?x ?y ?z)(when (cube1 ?x ?y ?z)
  (and (not(cube5 ?x ?y ?z)) (cube2 ?y ?x ?z
      ))))
(forall(?x ?y ?z)(when (cube3 ?x ?y ?z)
  (and (not(cube3 ?x ?y ?z)) (cube1 ?y ?x ?z
      ))))
(forall(?x ?y ?z)(when (cube4 ?x ?y ?z)
  (and (not(cube4 ?x ?y ?z)) (cube3 ?y ?x ?z
      ))))
(forall(?x ?y ?z)(when (cube2 ?x ?y ?z)
  (and (not(cube2 ?x ?y ?z)) (cube4 ?y ?x ?z
      ))))
;for edge cubelets
(forall(?x ?z)(when (edge13 ?x ?z)
  (and (not(edge13 ?x ?z)) (edge12 ?x ?z))))
(forall(?y ?z)(when (edge34 ?y ?z)
  (and (not(edge34 ?y ?z)) (edge13 ?y ?z))))
(forall(?x ?z)(when (edge24 ?x ?z)
  (and (not(edge24 ?x ?z)) (edge34 ?x ?z))))
(forall(?y ?z)(when (edge12 ?y ?z)
  (and (not(edge12 ?y ?z)) (edge24 ?y ?z))))
      ))
```

Listing 1: Action L of Rubik's Cube modeled in PDDL

- The action 'L', shown in Listing 1, is defined for a 90-degree clockwise rotation of the left face of the Rubik's Cube.
- The rotation results in a circular shift of colors towards the right on specific cube pieces, affecting only the colors along the X-axis and Y-axis due to the orientation of the left face in the Z-plane.
- Colors on corner pieces labeled as cube1, cube2, cube4, and cube3 are shifted circularly in sequence. Similarly colors on the edge pieces, edge12, edge13, edge24, edge34, also follow the circular shift resulting in 90-degree shift in colors as the action effect.

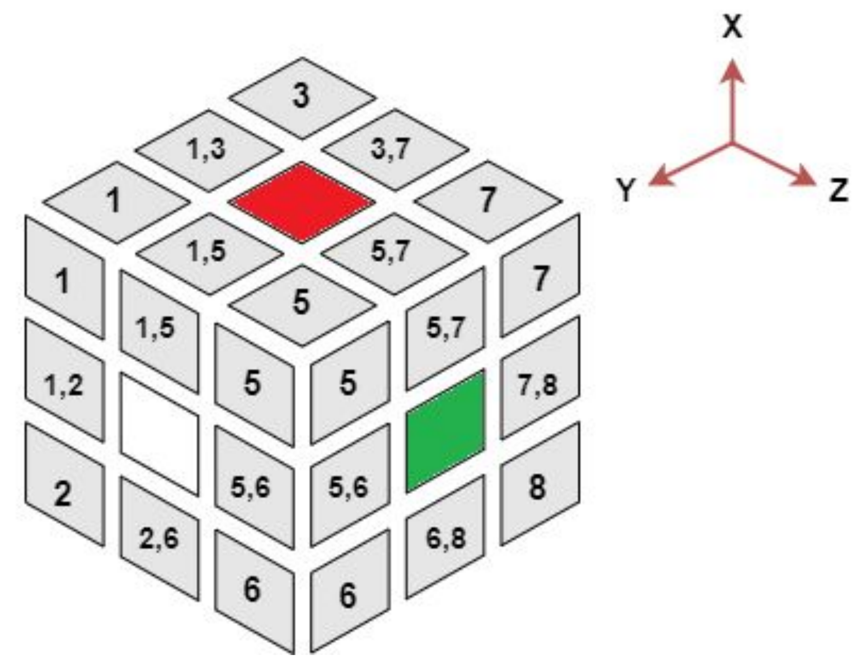

Figure-2: Rubik's cube description to define the domain encoding. For example, cube pieces 5, 6, 7, and 8 are in the right face

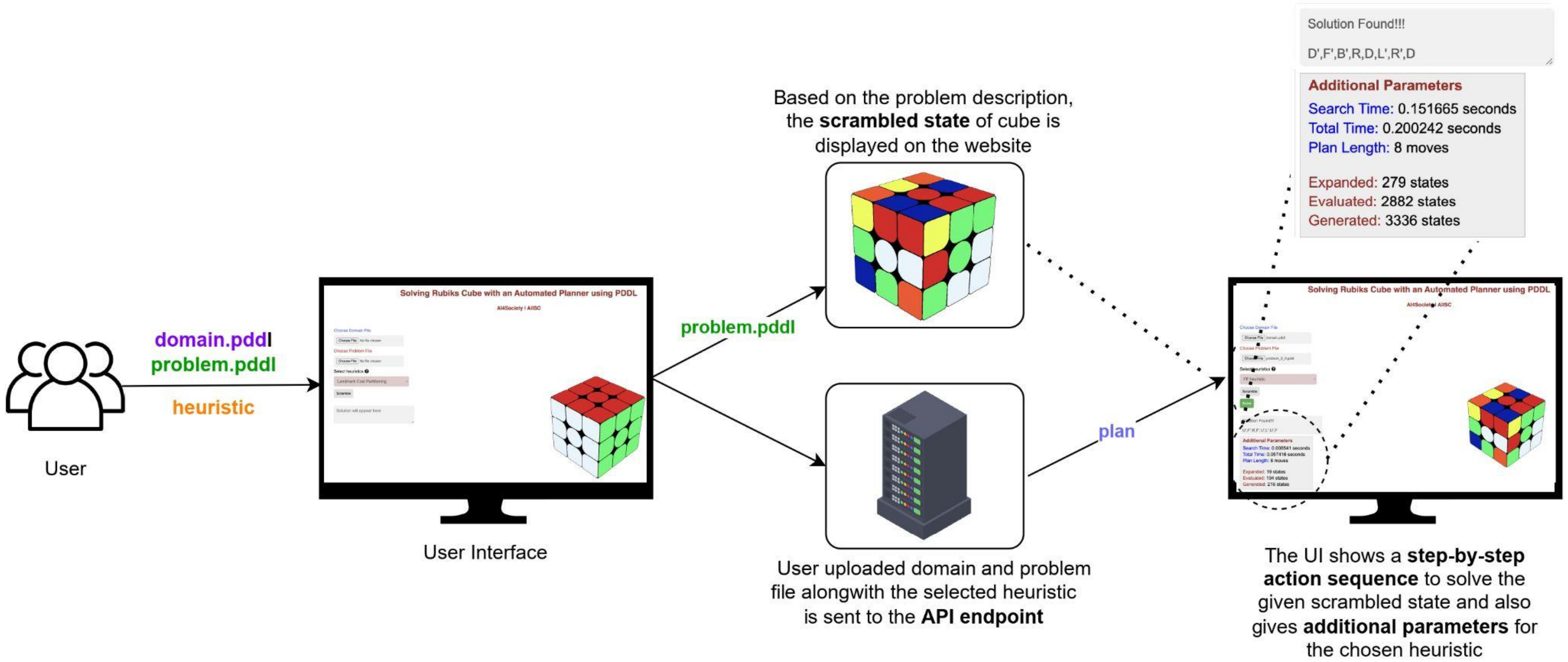

Figure-3: System Architecture of an Automated Rubik's Cube Solver: This diagram illustrates the workflow from user input through a web interface, where domain and problem files are uploaded and a heuristic is selected, to the backend server processing using PDDL, resulting in a displayed solution and performance metrics on the user interface.