CAB203 project

Dane Mckillop, n101960505

June 04, 2021

1. Introduction

The 2×2 Rubik's cube is a small puzzle with six faces, each face having four squares. In a completed state each face respectively is coloured red, blue, white, yellow, orange or green. Operations may be performed on the cube by rotating one of the faces of the cube by 90°, either clockwise or anti-clockwise. These rotations result in a distinct permutation, in which each of the six faces will have some combination of red, blue, white, yellow, orange or green squares.

Typically, the cube will start in a scrambled state, with the goal to perform operations so that the cube is solved, ending in the aforementioned completed state. Six sets of operations, excluding equivalent moves, may be performed to reach a new permutation of the cube. These are:

-A right face clockwise turn.
-A right face counter-clockwise turn.
-A front face clockwise turn.
-A front face counter-clockwise turn.
-An upper face clockwise turn.
-An upper face counter-clockwise turn.

Left, bottom and rear rotations will be omitted to reduce complexity. For example, a left face clockwise turn and a right face counter-clockwise turn result in the same permutation. Given any initial state of the cube, it is desirable to know the shortest number of rotations required to reach a completed state.

2. Instance Model

An instance of the problem is described by a set of faces, a starting state of the cube and a solved state of the cube. A state represents one of the 3,674,160 permutations of the cube which are reachable through legal moves. If this is the number of vertices in V, then it is not efficient to populate the set of vertices before operating. Additionally, we can derive a completed state of the cube as set of tuples whose members contain consistent elements. The example starting state and completed states of the cube provided, as well as a mapping of the cube are

$$\big((y,w,r,g),(b,r,y,r),(r,w,o,w),(o,y,g,y),(b,o,b,b),(g,g,o,w)\big)$$

$$\{(r,r,r,r),(b,b,b,b),(w,w,w,w),(y,y,y,y),(o,o,o,o),(g,g,g,g)\}$$

$$\begin{array}{cccc} & F_2 & & \\ F_0 & F_1 & F_4 & F_5 \\ & F_3 & & \end{array}$$

The instance is modelled by (s, d) where s is the starting state of the cube and d is the completed state of the cube. Then the instance model is

$$\begin{pmatrix} \big((y, w, r, g), (b, r, y, r), (r, w, o, w), (o, y, g, y), (b, o, b, b), (g, g, o, w)\big), \\ \{(r, r, r, r), (b, b, b, b), (w, w, w, w), (y, y, y, y), (o, o, o, o), (g, g, g, g)\} \end{pmatrix}$$

Regarding the completed combination d, it should be noted there are 24 orientations from which a completed cube may be viewed from. Representing these states through tuples is tedious, so d shall be represented here as a set. This is to emphasize that the combination of d's faces is irrelevant, only that they are correctly colour matched. For all other permutations, order of the faces is significant.

3. Solution Model

In order to find the minimum number of operations required to solve the cube, we need to identify the distance from our initial state to our completed state. As we have no predefined set of permutations, we shall record the permutations we arrive upon as we perform turns on the cube. A sequence of turns should not exceed 14, with the last turn ending in the completed state.

Therefore, the solution will be the length of the shortest sequence of turns between two states of the cube $s, a, b, c, \dots, d$ where the first state is the initial state and the last state is the completed state. For the purposes of this solution, the sequence of turns itself is not significant.

4. Problem Model

Let an instance (s, d) be given. We model the permutations of the cube as a graph $G = (V, E)$. The set of vertices V is the set of all $3{,}674{,}160$ permutations of the cube. That is,

$$\boldsymbol{V} = \left\{ \sum_{j=1}^{3{,}674{,}160} v_j \right\}$$

The set of edges E contains each 90° turn (u, v) between the vertices of state $u \in V$ distance $j - 1$ and permutation $v \in V$ distance j. Since each state has six edges connecting to six permutations that is,

$$\boldsymbol{E} = \left\{ \{u \in S\}: \sum_{j=1}^{n} (u_{j-1}, v_j) \right\}$$

Note: Unsure how to correctly calculate the value of *n* when accounting for cycles and irreflexivity. Initially, had 6(3,674,160).

We define $S \subseteq V$ since set V is unknown except for $s \in V$ and $d \in V$. Set S contains the vertices $u \in S$ distance j traversed to visit $d \in S$. That is,

$$\boldsymbol{S} = \begin{Bmatrix} \{s\} & : j = 0 \\ \{s,\ u_1,\ u_2, \ldots,\ u_{j-1}, d_j\} & : j \geq 1 \end{Bmatrix}$$

As the sequence of turns between s and d is not significant, the set of edges $T \subseteq E$ is not significant. For consistency, let us define the subset of edges $(u, v) \in T$ as the set of edges between vertices in set S, such that $T = S \times S$.

We can get from s to d by following a path in the subgraph $G' = (S, T)$ from s to d. Therefore, the solution is the length of the shortest path between s and d, given by the distance j at d starting from s.

5. Solution Model

As we are attempting to find the length of the shortest path from s to d, we shall implement a *breadth first search*. Each u vertex of a cube's state shall be a tuple of six tuples, each with four elements. As we do not know the u vertices of V, we define $S \subseteq V$, which will be required to track vertices in distance classes D_j.

To find each set D_j, we employ a recursive definition which will permute each $u \in D_{j-1}$ into a $v \in D_j$. Initially, $S = \{s\}$ and $D_0 = \{s\}$.

From here we recursively generate $6 \times v \in D_j$ from $u \in D_{j-1}$ by rearranging elements of tuple u, consistent with the physical equivalent of turning a face. From here, we check that $\forall v \in D_j$ are indeed j distance from s by set difference to see if $\forall v \in D_j : v \notin S$, such that

$$\boldsymbol{D_j} = D_j - S$$

We can also define $v \in D_j$ as being in the neighbourhood of D_{j-1} but not vertices in sets $D_0 \ldots D_{j-1}$, notated by

$$\boldsymbol{D_j} = N_{V_j}(D_{j-1}).$$

C shall be a set containing 24 vertices equivalent to d where order is significant. By intersection, we then evaluate if the cube has been solved where $\exists d \in D_j : d \in C$, that is

$$\{d\} = D_j \cap C$$

If the intersection is an empty set, S records the new visited vertices distance j from s for the next recursion, through a union of the neighbourhood of S_{j-1} with D_j**,** consistent with

$$\boldsymbol{S_j} = N_{V_j}(S_{j-1}) \cup D_j$$

If the intersection is a non-empty set, we have traversed shortest path length j of graph G' from s to d. There may be more than one d for each s, that is there may be multiple d vertices an equal distance j from vertex s. However, we only require j, so this is not significant. For any vertex $u \in D_j$, we can identify the length j of the shortest path to d as $\ell(u, d)$, therefore:

$$j(u) = \ell(u, d)$$

To summarise, the solution can be found from an instance as follows:

1. Initialize S and D to $\{s\}$.
2. Calculate D_j recursively.
3. Check if intersection of D_j and C is non-empty, return j if so.
4. Populate S by performing a union with D_j after each check.