THE COSET METHOD FOR RUBIK'S CUBE

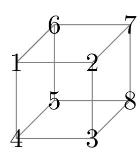
What's the maximum number of moves it takes to solve any position of Rubik's Cube? A basic but key idea used to answer this question is the coset method, which exploits the group structure of the Cube. In this handout, you will explore the coset method for a simplified version of the Cube.

1. A toy problem

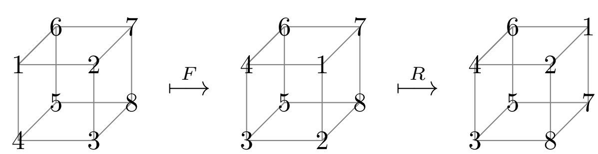
1.1. **Rubik's Cube basics.** Rubik's Cube consists of 8 corner pieces and 12 edge pieces around fixed axes joining the 6 center pieces. Every sequence of moves "mixes around" the corners among themselves, and the edges among themselves. For each type of piece, we distinguish two types of "mixings": a corner piece, for example, can be *oriented*, meaning twisted in place, either clockwise or counterclockwise; or it can be *permuted*, meaning moved to a different corner position.[1]

1.2. **A permutation puzzle.** In this problem, we consider a simplified version of the Cube where you only remember the permutation of the corners.

Here's one way to model this puzzle: take the Cube, remove all its stickers, and fix a numbering of the corners by $\{1, \ldots, 8\}$:



(If you prefer, you can imagine putting a sticker labeled i on all three sides of corner piece i.) Turning a face (see §2.1 for the notation) permutes the 8 corners:



Given any configuration of the 8 corners, the goal is to put them back into the initial configuration above.

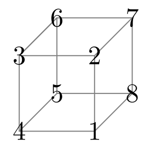
Date: April 19, 2017.

[1]Note that permutation is only defined after forgetting orientation. In other words, there is a short exact sequence of groups

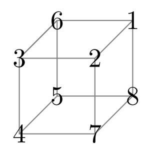
$$1 \to \{\text{orientations}\} \to \{\text{all "mixings"}\} \to \{\text{permutations}\} \to 1.$$

1.3. **Subgroup generated by half turns.** We can now state the first question.

> **Problem 1:** Given any state of this puzzle, how can you quickly tell if it can be solved using only half (180°) turns? (You're not allowed to change the orientation of the entire cube.)
>
> For example, can the following state be solved using only half turns?



> How about this one?



Here's a way to rephrase this more mathematically. Identify the states of this puzzle with S_8, the symmetric group on $\{1, \ldots, 8\}$. (To fix convention, we read composition in S_8 from left to right, and view it as acting on the set of puzzle states on the right. For details in the analogous setting of Rubik's Cube, see §2.1.) Let D be the subgroup generated by half turns.

> **Problem 1':** Give a "nice" characterization of D as a subgroup of S_8.
>
> What's the order of D?

Here, "nice" just means useful enough to answer Problem 1.

1.4. **The coset method.** The language of groups allows us to ask the following:

> **Problem 2:** Given any state of this puzzle, how can you quickly tell which right coset of D it's in?
>
> More precisely, find some enumeration (injection) $c\colon D\backslash S_8 \to \mathbb{N}$ of the coset space, so that given any puzzle state, corresponding to say $\sigma \in S_8$, you can quickly compute $c(D\sigma)$.

In plain words, c is a naming scheme for the cosets for D. Different cosets should have different names, and given any puzzle state, it should be easy to figure out the name of the coset containing that state. From this point of view, Problem 1 is the special case of the trival coset D.

You shouldn't work on the next problem unless you have nothing else to do.

> **Problem 3:** Implement the coset method for this puzzle using the subgroup D.

2. Background on the coset method

The problems above are a toy model for the coset method for solving Rubik's Cube.

2.1. **More Rubik's Cube basics.** Let X be the set of (legal) states of the Cube. Let $x_0 \in X$ be the solved state. Let G be the group of (legal) "mixings" of the Cube, where the group operation is composition; here, we only care about the effect on the Cube, not the actual sequence of moves. This group is generated by clockwise $90°$ turns of the six sides:

$$G = \langle U, D, R, L, F, B \rangle = \langle \text{Up, Down, Right, Left, Front, Back} \rangle.$$

As is conventional in the cubing community, we read composition from left to right: FR means apply F, then apply R. In other words, we view G as acting on X on the *right*. For example,

$$x_0.(FR) = (x_0.F).R$$

denotes the state obtained by applying F, then R, to the solved state. This gives an identification

$$\begin{aligned} X &\xleftarrow{\sim} G : \phi \\ x_0.g &\longleftarrow\!\shortmid\, g. \end{aligned} \tag{2.1}$$

Note that this is G-equivariant, where the right action of G on itself is by right multiplication:

$$\phi(g).g' = \phi(gg') \quad \text{ for all } g, g' \in G.$$

2.2. **Rubik's Cube as a graph search problem.** The goal is to describe an algorithm that finds a short solution to any given cube state in a reasonable time.

We want to rephrase this as a graph search problem. In general, given a set X with a right action (following our convention) of a group G and a set of generators $S \subset G$, define a labeled oriented graph $\Gamma(X, S)$ as follows: the vertices are identified with X, and there is an edge $x \xrightarrow{s^{\pm 1}} x'$ whenever $x.s^{\pm 1} = x'$ for $x, x' \in X$ and $s \in S$, i.e. a generator s or its inverse takes x to x'.

Now let X and G be as in §2.1. For Rubik's Cube, one common convention is to include half turns as generators:

$$S = \{U, D, R, L, F, B, U^2, D^2, R^2, L^2, F^2, B^2\} \subset G.$$

In terms of $\Gamma(X, S)$, finding a short solution for a given state x amounts to finding a short oriented path from x to x_0. This is a well-known problem with many efficient "informed graph search" algorithms (e.g. the A* algorithm). The details won't concern us; all we'll note is they require an appropriate "heuristic function" that estimates the distance from any given state to the goal state.

Remark 2.1. Another, more natural choice of generating set for G is

$$S' = \{U, D, R, L, F, B\} \subset G.$$

In the Rubik's Cube community, the length of a path in $\Gamma(X, S')$ is called the *quarter-turn metric* (QTM), while the length in $\Gamma(X, S)$ is called the *half-turn metric* (HTM). Since $\Gamma(X, S)$ has the same vertices as $\Gamma(X, S')$ but more edges, a state may have a shorter solution in HTM than in QTM.

Any pathfinding algorithm takes longer on a larger graph. The problem with pathfinding for Rubik's Cube is that $\Gamma(X, S)$ is huge! The number of vertices is the number of Cube states, or over 43 quintillion. To find a solution in a reasonable time, we need to be more clever.

2.3. **The coset method.** Let

$$T = \{U, D, R^2, L^2, F^2, B^2\} \subset G,$$

and let $Y \subset X$ be the subset of states that can be solved using only moves in T. To reduce computation, we split our pathfinding problem into two steps:

(1) Given any state, find a path to some state $y \in Y$.
(2) Find a path from y to the solved state that uses only half turns.

Step 2 is pathfinding on the smaller graph $\Gamma(Y, T)$, which requires less computation. What's not as clear is whether Step 1 is actually easier than the original problem. As a pathfinding problem on $\Gamma(X, S)$, there are a lot more goal states now (any state in Y), but an informed search algorithm requires a heuristic function. How would one quickly estimate how far a given state is from being in Y?

The key idea is to use the fact G that is a group. First of all, the G-equivariant bijection (2.1) induces an isomorphism

$$\Gamma(X, S) \cong \Gamma(G, S)$$

of labeled oriented graphs, so we may instead work with $\Gamma(G, S)$ (the Cayley graph), where the right action is by right multiplication. Consider the subgroup H generated by T:

$$H = \langle U, D, R^2, L^2, F^2, B^2 \rangle < G.$$

The identification (2.1) restricts to

$$\begin{aligned} Y &\xleftarrow{\sim} H \\ x_0.h &\longmapsfrom h, \end{aligned}$$

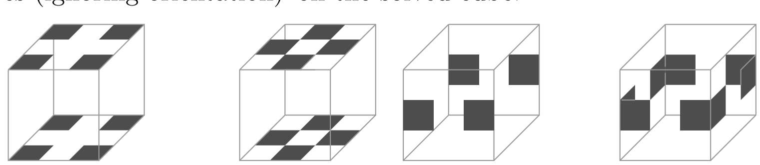
so Step 2 may be viewed as a pathfinding problem on $\Gamma(H, T)$.

Here's the punchline: G still acts on the right on the coset space $H\backslash G$. We can therefore view Step 1 as pathfinding on $\Gamma(H\backslash G, S)$, with goal state the trivial coset H. To be able to work efficiently with this graph, we need the following:

> **Key Task:** Given any Cube state, how can you quickly tell which right coset for H it's in?
>
> More precisely, find some enumeration (injection) $c\colon H\backslash G \to \mathbb{N}$ of the coset space, so that given any state, corresponding to say $g \in G$, you can quickly compute $c(Hg)$.

For the permutation puzzle of §1, this is precisely Problem 2.

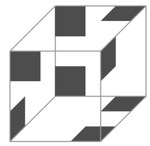
Here is one way to do this for $H\backslash G$. Consider the following three collections–of corner stickers, of edge stickers (drawn on two separate cubes for clarity), and of edge pieces (ignoring orientation)–on the solved cube:



corner stickers edge stickers edge pieces

Note that the action of H on X preserves each collection (ignoring any mixing within each collection), so it makes sense to talk about the effect of a coset Hg on these collections. In fact, H is exactly the subgroup of G preserving these collections, so these effects uniquely determine Hg.

This can be turned into a naming scheme for the cosets. For example, on a given cube state, the first collection might look like this:



In each corner position, the marked sticker can be oriented from the marked sticker in the solved state in three ways: call these 0, 1, and 2. Going through all corner positions in some fixed order, we get an 8-digit sequence in $\{0, 1, 2\}$ that serves as a "tag" recording the effect of this state on the first collection. Using the same corner numbering scheme from §1, the example above might produce the tag

$$11011200.$$

A similar procedure produces a tag from each of the other two collections, and we can take these three tags together as the name of the coset containing this state. It's straightforward to turn this into an enumeration $c\colon H\backslash G \to \mathbb{N}$.

2.4. **A little history.** The use of the coset method for Rubik's Cube goes back to M. Thistlethwaite, who used it in 1981 to prove an upper bound of 52 on the diameter of the Cube group (in HTM, see Remark 2.1). The Thistlethwaite algorithm has 4 steps, corresponding to a chain of subgroups of the same length:

$$1 = G_4 < G_3 < G_2 < G_1 < G_0 = G.$$

(See for example [Sch].)

What we described in this section is the very basics of H. Kociemba's "two-phase algorithm" developed in 1991 and 1992 [Koc]. The subgroup H is Thistlethwaite's G_2, so this algorithm combines the first 2 steps and the last 2 steps of Thistlethwaite's. In Kociemba's terminology, the Key Task above is to find a "coset coordinate" for $H\backslash G$.

In 1995, Michael Reid reduced the upper bound for the diameter of the Cube group to 29 by analyzing Kociemba's algorithm, and also proved a lower bound of 20. In 2010, T. Rokicki, H. Kociemba, M. Davidson, and J. Dethridge reduced the lower bound to 20, thus proving that the maximum number of moves it takes to solve any position of Rubik's Cube is 20 [RKDD].

References

[Koc] Herbert Kociemba. "Solve Rubik's Cube with Cube Explorer". `http://kociemba.org/cube.htm`.

[RKDD] Tomas Rokicki, Herbert Kociemba, Morley Davidson, and John Dethridge. "God's Number is 20". `http://cube20.org/`.

[Sch] Jaap Scherphuis. "Computer Puzzling". `http://www.jaapsch.net/puzzles/compcube.htm#kocal`.