Algorithms for Solving Rubik's Cubes

Dmytro Pashchenko [1]

Abstract

This is a short summary of a paper by Demaine et al. under the same title [1]. The Rubik's Cube, perhaps the most popular puzzle in the world, has a rich underlying algorithmic structure. The authors of the presented article prove that the diameter of the configuration space for $n \times n \times 1$ and $n \times n \times n$ Rubik's Cubes is of $\Theta(n^2/\log n)$. This leads them to an asymptotically optimal algorithm for solving a general Rubik's Cube in the worst case. Additionally, Demaine et al. show that the problem of finding the optimal solution becomes NP-hard in an $n \times n \times 1$ Rubik's Cube, and speculate how an $O(1) \times O(1) \times n$ Cube can be solved in a polynomial time.

1. Introduction

To a theoretical computer scientist, the Rubik's Cube is a source of numerous open questions. What are good algorithms for solving a Rubik's Cube puzzle? What is an optimal bound on the number of moves in the worst case? What is the complexity of optimizing the number of moves, given a certain starting configuration? How do the side lengths of the cube affect the worst-case complexity?

Although it is already known that the God's Number (diameter of the configuration space) is 20 for the $3 \times 3 \times 3$ Rubik's Cubes, the optimal solution for each configuration is only yet to be found [2]. The authors of the original paper confirm that the general approach for solving the Rubik's Cubes can be generalized to have an upper bound of $O(n^2)$ for an arbitrary $n \times n \times n$ cube. Moreover, the solution can be optimized using parallelism with the number of moves reduced to $\Theta(n^2/\log n)$.

Additionally, there is no answer [3], whether it is NP-hard to solve a given $n \times n \times n$ or $n \times n \times 1$ Rubik's Cube using the fewest possible moves. On the positive side, Demaine et al. show how to compute the exact optimum for $O(1) \times O(1) \times n$ Rubik's Cubes. On the negative side, they prove that it is NP-hard to find an optimal solution to a subset of cubies in an $n \times n \times 1$ Rubik's Cube.

[1] Institute of Computer Science, University of Tartu, Tartu, Estonia, dmytro.pashchenko@ut.ee.

2. Diameters of general Rubik's Cubes

2.1. Diameter of $n \times n \times 1$ Rubik's Cube

For simplicity, the edge and corner cubies were ignored and it was assumed that n is even. The third coordinate of a cubie was omitted since it is always 0. A cube is said to be solved if the top of the cube is orange. *Cubie cluster* (x, y) is a set of locations reachable by a cubie at position (x, y). By flipping rows and columns, we can get four reachable locations for a cubie (x, y) that form a cluster: (x, y), $(n-x-1, y)$, $(x, n-y-1)$, $(n-x-1, n-y-1)$.

First of all, the authors showed than any reachable cluster configuration can be solved with a sequence of moves of constant length without affecting other clusters. Figure 1 gives such a sequence for each possible cluster configuration. Here, the row and column moves are denoted by H_i and V_i respectively.

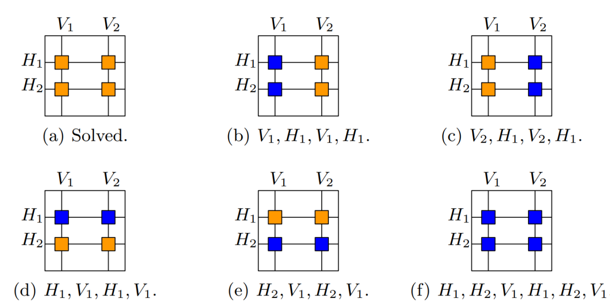

Figure 1. The reachable cluster configurations and the move sequences to solve them.

Given that there are n^2 clusters in $n \times n \times 1$ Rubik's Cube, we can solve the entire cube with a sequence of $O(n^2)$ moves by using the sequences from Figure 1 to solve each cluster individually. Suppose that we are given rows X and columns Y such that all the clusters $(x, y) \in X \times Y$ are in the same configuration. Then, for solving all the clusters individually, the number of moves required is $\Theta(|X| \cdot |Y|)$.

Instead of repeating the same row/column move for each cubie in a row/column of the same cluster type, we can do it only once. Thus, the cubie clusters $(x, y) \in X \times Y$ all can be solved using only $O(|X| + |Y|)$ moves, if each one of those clusters is in the same configuration.

This technique was used to solve all cubie clusters in a particular cluster configuration without affecting other clusters (see the full article [1]). Constructing the move sequence for all 6 configurations and taking the advantage of parallelism, the authors got the following result:

Theorem 1 *Given an $n \times n \times 1$ Rubik's Cube configuration, all cubie clusters can be solved in $O(n^2/\log n)$ moves.*

Theorem 2 *Some configurations of an $n \times n \times 1$ Rubik's Cube are $\Omega(n^2/\log n)$ moves away from being solved.*

2.2. Diameter of $n \times n \times n$ Rubik's Cube

For simplicity, it was again assumed that n is even and all edge and corner cubies were ignored. Suppose that each cubie has a face coordinate $(x, y) \in \{0, 1, \dots, n-1\} \times \{0, 1, \dots, n-1\}$. Then for a cubie (x, y), a face rotation of the front face gives us four reachable locations: (x, y), $(n-x-1, y)$, $(x, n-y-1)$ and $(n-x-1, n-y-1)$. Row and column moves just move the cubie to another face where it takes one of those four locations. Therefore, it can reach 24 locations in total, which together form a *cubie cluster* for the cubie (x, y).

There are twelve types of *face moves*, two for each face. Demaine at al. also added a thirteenth type which applies the identity function. Given a face move type a, they wrote F_a to denote the move itself. Provided a particular index $i \in \{1, 2, \dots, \lfloor n/2 \rfloor - 1\}$, there are twelve types of *row and column moves* that can be performed: three different axes for the slice, two different indices (i and $n-i-1$) to pick from, and two directions of rotation. Similarly, the authors wrote $RC_{a,i}$ to denote the move itself, where a is the type of row or column move, and i is the index.

Again, suppose we have a set of columns $X = \{x_1, \dots, x_l\}$ and rows $Y = \{y_1, \dots, y_k\}$ such that $X \cap Y = \emptyset$ and all cubie clusters $(x, y) \in X \times Y$ are in the same cluster configuration d. Solving each cluster individually would require $\Theta(|X| \times |Y|)$ moves. Likewise, the authors attempted to parallelize the solution. For this, they constructed the following sequence of moves (solving configudation d) as a building block:

$$\mathrm{BULK}_i = F_{a_i}, RC_{b_i,x_1}, \dots, RC_{b_i,x_l}, RC_{c_i,y_1}, \dots, RC_{c_i,y_k}.$$

The full sequence $\mathrm{BULK}_1, \mathrm{BULK}_2, \dots, \mathrm{BULK}_m$ would solve all clusters $X \times Y$ in $O(|X| + |Y|)$ moves. The only other clusters it may affect are the clusters $X \times X$ and $Y \times Y$. After effectively solving the problem of affected clusters, Demaine et al. arrived at the same conclusion:

Theorem 3 *Given an $n \times n \times n$ Rubik's Cube configuration, all cubie clusters can be solved in $O(n^2/\log n)$ moves.*

Theorem 4 *Some configurations of an $n \times n \times n$ Rubik's Cube are $\Omega(n^2/\log n)$ moves away from being solved.*

3. Optimally Solving the Rubik's Cubes

3.1. Optimally Solving a Subset of the $n \times n \times 1$ Rubik's Cube is NP-Hard

Say that we are given a configuration of an $n \times n \times 1$ Rubik's Cube and a list of *important* cubies. The task is to find the shortest sequence of moves that solves the important cubies. An *ideal solution* for a subset of important cubies in a particular $n \times n \times 1$ puzzle is a solution that can solve all the cubies with the smallest possible number of moves. The authors showed that the problem of finding a subset of the cubies that has an ideal solution can be reduced to a Not-All-Equal 3-SAT problem, which is known to be NP-hard.

3.2. Optimally Solving an $O(1) \times O(1) \times n$ Rubik's Cube

Consider a $c_1 \times c_2 \times n$ Rubik's Cube. A slice (a set of cubies) is called *short* if the cubies match in z-coordinate; otherwise, a slice is called *long*. The pair of slices $z = i$ and $z = (n-1) - i$ form the i-th cubie cluster. Any short move affects the cubies in exactly one cubie cluster. Any sequence of long moves involves arranging $c_1 \cdot c_2$ blocks of cubies with dimensions $1 \times 1 \times n$. Each such arrangement is called a *long configuration*. The algorithm, proposed by Demaine et al., gradually solves each cluster by performing the short moves necessary for solving each cluster within the current long configuration. Thus, by repeatedly going through all possible long configurations and inserting short moves into the appropriate places, a full solution to the puzzle can be obtained in a polynomial time.

4. Conclusion

The diameter of the configuration space for $n \times n \times 1$ and $n \times n \times n$ Rubik's Cubes proved to be of $\Theta(n^2/\log n)$. This gave an asymptotically optimal algorithm for solving a general Rubik's Cube in the worst case. The authors justified NP-hardness of finding an optimal solution to a subset of cubies in $n \times n \times 1$ Rubik's Cube. Finally, they showed how to compute optimum for $O(1) \times O(1) \times n$ Rubik's Cubes.

References

[1] Erik D. Demaine, Martin L. Demaine, Sarah Eisenstat, Anna Lubiw, Andrew Winslow. Algorithms for Solving Rubik's Cubes, 2011.

[2] Tomas Rokicki, Herbert Kociemba, Morley Davidson, and John Dethridge. God's number is 20, 2010. http://cube20.org.

[3] Graham Kendall, Andrew Parkes, and Kristian Spoerer. A survey of np-complete puzzles. International Computer Games Association Journal, 31(1):13–34, 2008.