

**Rubic’s cube solver**

**Sanikommu Sri LakshmiReddy 11802922**

Table of Contents:

1. Overview of Project
2. Introduction

* Common Definitions
* Theorem

1. Related Work
2. Implementation
3. Reference

Overview of Project:

The project which we took up is Stock Market Prediction. In this project we are supposed to predict a company’s stock value for 1 month and 6 months.

Stock Market Prediction is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. Stock Market is the important part of economy of the country and plays a vital role in the growth of the industry and commerce of the country that eventually affects the economy of the country. Both investors and industry are involved in stock market and wants to know whether some stocks will rise or fall over a certain period. The stock market is the primary source of any company to raise funds for business expansion. It is based on the concept of demand and supply. If the demand for a company’s stock is higher, then the company share price increases and if the demand for company’s stock is low then the company share price decreases.

Introduction:

1 Introduction A little over thirty years ago, Hungarian architecture professor Ern˝o Rubik released his “Magic Cube” to the world.4 What we now all know as the Rubik’s Cube quickly became a sensation [26]. It is the best-selling puzzle ever, at over 350 million units [15]. It is a tribute to elegant design, being part of the permanent collection of the Museum of Modern Art in New York [18]. It is the heart of World Cube Association’s speed-cubing competitions, whose current record holders can solve a cube in under 7 seconds (or 31 seconds blindfold) [1]. Another puzzle that can be described as a permutation group given by generators corresponding to valid moves is the n × n generalization of the classic Fifteen Puzzle. This n 2 − 1 puzzle also has polynomial diameter, though lacking any form of parallelism, the diameter is simply Θ(n 3 ) [20]. Interestingly, however, computing the shortest solution from a given configuration of the puzzle is NP-complete [21]. More generally, given a set of generator permutations, it is PSPACE-complete to find the shortest sequence of generators whose product is a given target permutation [5,11].

2 Common Definitions

We begin with some terminology. An ` × m × n Rubik’s Cube is composed of `mn cubies, each of which has some position (x, y, z), where x ∈ {0, 1, . . . , `−1}, y ∈ {0, 1, . . . , m−1}, and z ∈ {0, 1, . . . , n−1}. Each cubie also has an orientation. Each cubie in a Rubik’s Cube has a color on each visible face. There are six colors in total. We say that a Rubik’s Cube is solved when each face of the cube is the same color, unique for each face. A slice of a Rubik’s Cube is a set of cubies that match in one coordinate (e.g. all of the cubies such that y = 1). A legal move on a Rubik’s Cube involves rotating one slice around its perpendicular5 . In order to preserve the shape of the cube, there are restrictions on how much the slice can be rotated. If the slice to be rotated is a square, then the slice can be rotated 90◦ in either direction. Otherwise, the slice can only be rotated by 180◦ . Finally, note that if one dimension of the cube has length 1, we disallow rotations of the only slice in that dimension. For example, we cannot rotate the slice z = 0 in the n × n × 1 cube.

**RELATED WORK:**

**4 Diameter of n × n × n Rubik’s Cube**

Because the only visible cubies on the n×n×n Rubik’s Cube are on the surface, we use an alternative coordinate system. Each cubie has a face coordinate (x, y) ∈ {0, 1, . . . , n − 1} × {0, 1, . . . , n − 1}. Consider the set of reachable locations for a cubie with coordinates (x, y) ∈ {0, 1, . . . , n − 1} × {0, 1, . . . , n − 1} on the front face. A face rotation of the front face will let it reach the coordinates (n−y−1, x), (n−x−1, n−y−1), and (y, n−x−1) on the front face. Row or column moves will allow the cubie to move to another face, where it still has to have one of those four coordinates. Hence, it can reach 24 locations in total. For this problem, we define the cubie cluster (x, y) to be those 24 positions that are reachable by the cubie (x, y). We define edge cubies to be cubies with more than one face visible. If a cluster has an edge cubie, then all of its cubies are edge cubies. We call such clusters edge clusters. We define corner cubies to be cubies with more than two faces visible. All corner cubies are in a single cluster known as the corner cluster. If n is odd, we must also define several other types of cubies. We first define cross cubies to be cubies with face coordinates of the form (x,(n − 1)/2) or ((n − 1)/2, y). If a cluster contains a cross cubie, then all of its cubies are cross cubies, and the cluster is called a cross cluster. We define center cubies to be the six cubies with face coordinates ((n − 1)/2,(n − 1)/2). They form a special cluster which we will call the center cluster. Our goal in solving the Rubik’s Cube will be to make each side match the color of its center cluster. Hence, there is no need to solve the center cluster A cluster move sequence consists of three type sequences: face types a1, . . . , a`, row and column types b1, . . . , b`, and row and column types c1, . . . , c`. For a cubie cluster (x, y), the sequence of actual moves produced by the cluster move sequence is Fa1 , RCb1,x, RCc1,y, . . . , Fa` , RCb`,x, RCc`,y. A cluster move solution for a cluster configuration d is a cluster move sequence with the following properties: 1. For any (x, y) ∈ {1, 2, . . . , bn/2c − 1} × {1, 2, . . . , bn/2c − 1}, if cluster (x, y) is in configuration d, then it can be solved using the sequence of moves Fa1 , RCb1,x, RCc1,y, . . . , Fa` , RCb`,x, RCc`,y. 2. The move sequence Fa1 , RCb1,x, RCc1,y, . . . , Fa` , RCb`,x, RCc`,y does not affect cubie cluster (y, x). 3. All three of the following sequences of moves do not affect the configuration of any cubie clusters: Fa1 , RCb1,x, Fa2 , RCb1,x, . . . , Fa` , RCb`,x; Fa1 , RCc1,y, Fa2 , RCc1,y, . . . , Fa` , RCc`,y; Fa1 , Fa2 , . . . , Fa` . Our goal is to construct a cluster move solution for each possible cubie cluster configuration, and then use the cluster move solution to solve multiple cubie clusters in parallel.

These cluster configurations enforce the following constraints:

I1(˜x1) < I1(˜y2) < I2(˜x1) < I2(˜y2), I1(˜x2) < I1(˜y3) < I2(˜x2) < I2(˜y3),

I1(˜y1) < I1(˜x1) < I2(˜y1) < I2(˜x1), I1(˜y2) < I1(˜x2) < I2(˜y2) < I2(˜x2).

We can use these inequalities to construct the following chains:

I1(˜y1) < I1(˜x1) < I1(˜y2) < I1(˜x2) and I2(˜y1) < I2(˜x1) < I2(˜y2) < I2(˜x2)

To ensure that cubie cluster (x1, y1) still remains solved, we need I1(x1) < I1(y1). Given the configuration of cubie cluster (x1, y3), we need I1(y3) < I1(x1). To ensure that cubie cluster (x3, y3) still remains solved, we need I1(x3) < I1(y3). Given the configuration of cubie cluster (x3, y1), we need I1(y1) < I1(x3). Thus I1(y3) < I1(x1) < I1(y1) < I1(x3) < I1(y3), a contradiction. Hence this case is also impossible. Because neither of the two cases are possible, I1(x2) must lie between I1(x1) and I1(x3), which is precisely what we wanted this gadget to enforce. We must also show that this gadget does not enforce any constraints other than the ones expressed in the lemma. Given any solution which satisfies the requirements given in the lemma, we must be able to insert the moves for the new rows and columns in such a way that all important clusters will be solved. The extra constraints on the existing move sequence ensure the following: max x∈{x1,x3} I1(x) < min x∈{x1,x3} I2(x). So we know that we can place the moves for the extra rows and columns used by the gadget from Lemma 13. We need only determine where to insert the moves for the three extra rows added by this gadget. The constraints given in the statement of the lemma allow for four different possible orderings of all of the x1, x2, x3 moves. We consider each case separately.

1. I1(x1) < I1(x2) < I1(x3) < I2(x2) < I2(x1) < I2(x3). Then we insert the moves y1, y2, y3 so that the following is a subsequence of the move sequence: y2, y3, x1, y1, x2, y3, x3, y1, x2, y2, x1, x3. 2.
2. I1(x1) < I1(x2) < I1(x3) < I2(x2) < I2(x3) < I2(x1). Then we insert the moves y1, y2, y3 so that the following is a subsequence of the move sequence:

y2, y3, x1, y1, x2, y3, x3, y1, x2, y2, x3, x1

3. I1(x3) < I1(x2) < I1(x1) < I2(x2) < I2(x1) < I2(x3). Then we insert the moves y1, y2, y3 so that the following is a subsequence of the move sequence: y1, y2, x3, y3, x2, y1, x1, y3, x2, y2, x1, x3

4. I1(x3) < I1(x2) < I1(x1) < I2(x2) < I2(x3) < I2(x1). Then we insert the moves y1, y2, y3 so that the following is a subsequence of the move sequence: y1, y2, x3, y3, x2, y1, x1, y3, x2, y2, x3, x1

Theorem

Given any c1 × c2 × n Rubik’s Cube configuration, it is possible to find the optimal solution in time polynomial in n. Proof. By Lemma 17, we know that the total number of long moves in the optimal solution is at most (c1c2)! · 2 1+3c1c2+8(c1+c2) , which is constant. We also know that there are a total of c1 + c2 possible long moves. Hence, the total number of possible sequences of long moves is constant, so we can enumerate all of these in time O(1). For each of the sequences of long moves, we want to find the optimal solution using that sequence of long moves. Because the long moves are fixed, we can calculate the short moves for each cubie cluster independently. To calculate the short moves for some fixed cubie cluster, we note that between two sequential long moves, there are at most four different ways to rotate each of the two slices in the cubie cluster, for a total of at most sixteen possible combinations of short moves. For a given cubie cluster, we have to consider ≤ 161+(c1c2)!·2 1+3c1c2+8(c1+c2) possible combinations. This is constant, so we can try all possibilities to see if they solve the cubie cluster. We can pick the shortest of those. If we perform this operation for all cubie clusters, we will have an optimal solution for this particular sequence of long moves. If we calculate this for all sequences of long moves, then we can pick the overall optimal solution by taking the sequence of minimum length.

IMPLEMENTATION:

Once we acquire a dataset, we intend to divide it into two subsets:

***Training set***: is a subset of the dataset used to build predictive models.

***Test set or unseen examples:*** is a subset of the dataset to assess the likely future performance of a model. If a model fit to the training set much better than it fits the test set, over fitting is probably the cause.

**IMPLEMENTATION:**

This is the code which we used for stock market prediction.

**CONCLUSION:**

In this paper, we presented several new results. First, we introduced a technique for parallelizing the solution to two types of generalized Rubik’s Cubes. As a result, we showed that the diameter of the configuration space for these two types of Rubik’s Cubes is Θ(n 2/ log n). In addition, we showed that it is NP-hard to find the shortest sequence of moves which solves a given subset of the cubies in an n×n×1 Rubik’s Cube. Finally, we showed that there exists a polynomialtime algorithm for solving Rubik’s Cubes with dimensions c1 × c2 × n, where c1 6= n 6= c2

**REFERENCES:**

* <https://ruwix.com/online-rubiks-cube-solver-program/>
* https://www.researchgate.net/publication/51913939