



**KTH Computer Science  
and Communication**

# **Benchmarking Human Solving Methods for Rubik's cube**

Duis autem vel eum iruire dolor in hendrerit in vulputate velit esse molestie consequat,  
vel illum dolore eu feugiat null

ANDREAS NILSSON ANIL9@KTH.SE  
ANTON SPÅNG ASPANG@KTH.SE

DD143X - Bachelor Thesis  
Supervisor: Michael Schliephake  
Examiner: Örjan Ekeberg

TRITA xxx yyyy-nn



# Abstract

This is a skeleton for KTH theses. More documentation regarding the KTH thesis class file can be found in the package documentation.

# Sammanfattning

Denna fil ger ett avhandlingsskelett. Mer information om  
L<sup>A</sup>T<sub>E</sub>X-mallen finns i dokumentationen till paketet.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problem Definition . . . . .	2
1.2	Problem Statement . . . . .	2
1.3	Purpose . . . . .	2
1.4	Structure . . . . .	2
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	Competitions . . . . .	3
2.1.1	Speedcubing . . . . .	3
2.1.2	Fewest moves . . . . .	3
2.2	Rubik's Cube . . . . .	3
2.2.1	Description . . . . .	3
2.2.2	Notation . . . . .	4
2.3	Algorithms . . . . .	4
2.3.1	Layer-by-layer using daisy method . . . . .	5
2.3.2	Dedmore algorithm . . . . .	7
<b>3</b>	<b>Method</b>	<b>9</b>
3.1	Literature study . . . . .	9
3.2	Implementation and data collection . . . . .	9
3.3	Analyze and representation . . . . .	9
<b>4</b>	<b>Implementation</b>	<b>11</b>
4.1	Cube representation . . . . .	11
4.2	Algorithms . . . . .	11
4.3	Scramble . . . . .	11
4.4	Difficulty . . . . .	11
<b>5</b>	<b>Results and Analyze</b>	<b>13</b>
5.1	Data . . . . .	13
5.2	Comparison . . . . .	13
<b>6</b>	<b>Discussion</b>	<b>15</b>
6.1	Comparison . . . . .	15

6.2 Errors . . . . .	15
<b>7 Conclusion</b>	<b>17</b>
<b>References</b>	<b>19</b>
<b>Bilagor</b>	<b>19</b>
<b>A Calculation of randomly rotate faces</b>	<b>21</b>

# Chapter 1

## Introduction

The Rubik's cube is an 3-D combination puzzle, where each side of the cube is covered with nine squares in six possible colours: white, red, blue, orange, green and yellow.

You start with a scrambled cube, meaning that the colored miniature cubes (cubies) are randomly positioned by using random operations on the cube ( fig 1.1a) .

The goal is to obtain the unique solution where each side of the cube are covered with only one colour per side (fig 1.1b). Different methods have been developed with series of notation to solve subproblems one at the time to reach the unique solution. The general idea which many methods are based on is to solve it one layer at the time.

If you were to randomly rotate the faces in an attempt to solve the cube, there is almost zero chance of achieving the solved state in your lifetime because of all the possible permutations of the cube. There are  $4.3 * 10^{19}$  (or 43 quintillion) [3] different states. Assuming you get to a unique state every second it would take more than 130 billion years to test 10% of the cubes possible states [appendix A].

There are two major ways to compete in solving the Rubik's cube: the least amount of moves and solving the cube as fast as possible (speedcubing). The cube was invented by the professor of architecture Ernő Rubik as a teaching tool to help his student understand 3D objects. It was not until he scrambled his new cube and

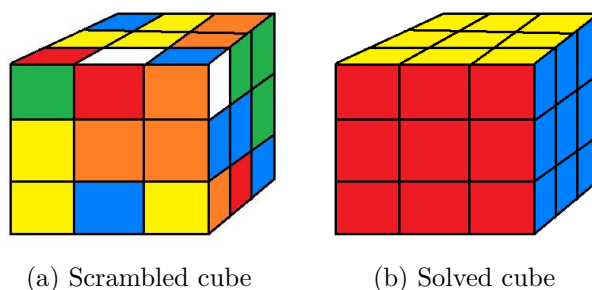


Figure 1.1

tried to restore it, that he realize that his creation was a puzzle.  
The cube was originally called magic cube and was licensed to be sold by the american toy company Ideal Toy Company in 1980. [5].

## 1.1 Problem Definition

This thesis will explore two commonly known beginner methods for human solving of rubik's cubes. Both based on the idea to solve the cube one layer at the time which is the easiest way for the human mind to solve this kind of problem [4]. We will evaluate both methods operation variance, number of operations and time to solve, to find the most time-efficient and move-efficient of them both.

During the literature study there was no such comparison of solving-methods for 3x3x3 cube found. The information found about solving-methods for 3x3x3 cubes was tutorials of how to use them..

## 1.2 Problem Statement

Which of the beginner algorithms would be more effective for speedcubing?  
Which of the beginner algorithms solves the cube with the least amount of moves?  
Which beginner algorithm is easiest to learn and execute to someone inexperienced with the cube?

## 1.3 Purpose

The purpose is to test two methods for solving rubik's cube to find out which uses the least amount of moves and if the same method is less time-consuming than the other. The inexperienced user will find this as a guideline as to which algorithm to start his/her journey towards solving the Rubik's cube.

## 1.4 Structure

The second section will introduce the reader to concepts necessary to understand the algorithms implemented and benchmarked. The third section will explain the methods used in this thesis. The fourth sections will explain the implementations made in detail and the difficulties.

The fifth section will be for presenting the results and the sixth section for discussion regarding the results. After that will there be a conclusions section that completes the circle of the thesis, answering the problem statements. Lastly the references used to this thesis will be listed followed by appendix containing computations and graphs.



## Chapter 2

# Background

Providing the reader with information of how the cube works and operations that you can perform on it as well as the algorithms that are in focus for this thesis.

### 2.1 Competitions

There are two types of competitions regarding the cube.

#### 2.1.1 Speedcubing

When competing in an official event regulated by the World Cube Association (WCA), the competitor has at maximum 15 seconds of inspection time of the cube before the solve begins.[7] The time stops when the competitor have reached the unique solution.

#### 2.1.2 Fewest moves

The competitors have 60 minutes without any inspection time and the competitor should also be able to hand in a written solution with the notations used in the correct format [7].

### 2.2 Rubik's Cube

Explanation of how the cube is constructed and the different notations for the operations.

#### 2.2.1 Description

The cube consists of 26 cubies with three,two or one visible sides depending on the type of cubie. There is one core piece consisting of three axes which holds the center pieces together [6]. The corner and edge pieces (fig 2.1) are the cubies that

are movable to different edge and corner positions, the center pieces (fig 2.1) can only be moved according to the axis.

### 2.2.2 Notation

The notation describes the different move operations on the cube. This thesis uses the notation used by WCA. Explained below:

Clockwise 90 degrees:

F - Front face

B - Back face

R - Right face

L - Left face

U - Upper face

D - Bottom face

To denote the anti-clockwise 90 degrees rotation just put a single citation mark (‘) after the letter. For example F’ - move front face anti-clockwise 90 degrees.[8] To denote clockwise 180 degrees rotation just put (2) after the letters described above.

## 2.3 Algorithms

The algorithms are physically methods for solving the cube by hand but matches the mathematical definition of an algorithm. Unlike many of the “near-optimal solvers” (some of simulates several moves ahead to find the optimal move[2]), these algorithms can with little effort be taught to humans.

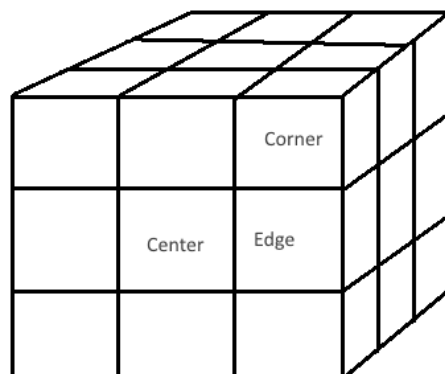


Figure 2.1: Cubie types

## 2.3. ALGORITHMS

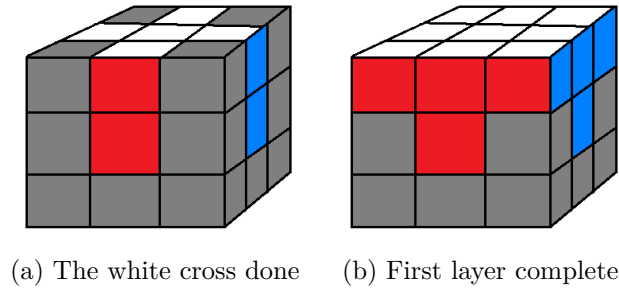


Figure 2.2

### 2.3.1 Layer-by-layer using daisy method

The layer-by-layer (LBL) algorithms divides the cube into layers and makes it possible to solve the subproblems without breaking any pieces already made. The daisy method is to solve the white cross by first make a cross with white edges and a yellow center and then turn the white edges, completing the white cross in the bottom [1].

#### 1. White cross

The goal here is to achieve a white cross, so that the white center-piece is aligned with its 4 white edge-pieces in the bottom. For it to be a completed step, the second color of the white edge-pieces must also align with it's center-piece counterpart as shown in fig 2.2a. This is done by first making a cross on the top with the yellow center-piece and then flip the edges over to the bottom one at the time using an operation, to make sure the edge-pieces on the vertical sides are aligned with the center-pieces.

#### 2. White corners

With the white cross done the next step is to complete the first layer by positioning the white corner-pieces correct between the cross edges (fig 2.2b). This is achieved with three different operation-combinations depending on the colour positions, all focusing on the upper-front-left corner of the cube.

#### 3. Middle layer edges

The next step is to solve the middle layer by moving down a cubie from the middle of the third layer to the correct position on the second layer (fig 2.3a).

#### 4. Yellow cross

With the second layer complete (fig 2.3b) it is time to work on the yellow cross. This is achieved by applying one of two operation-combinations or both depending

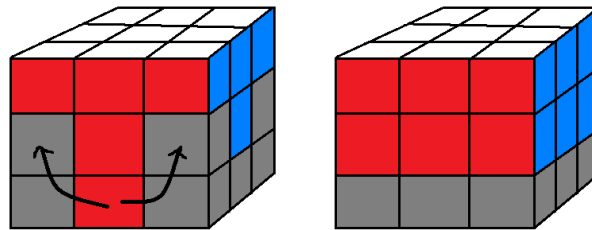
on how many white pieces that are correct positioned (fig 2.4).

### 5. Yellow corners

Positioning the yellow corners by applying different operations to move the edges depending on how many corners that are in position already.

### 6. Last layer permutation

Now when the corners of the last layer are in position so that the top is all yellow, the only thing left to do is to positioning the pieces of the last layer so they match with the colours of the vertical sides.



(a) Technique for each side (b) Middle layer complete

Figure 2.3

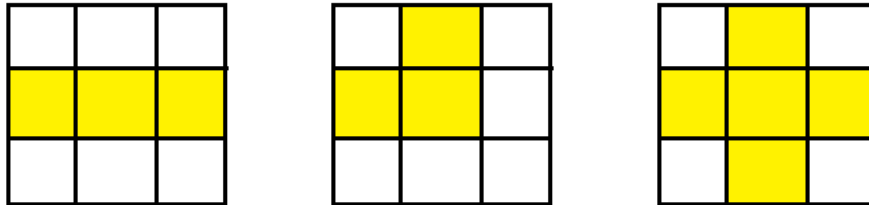


Figure 2.4: Different states to achieve the yellow cross

## 2.3. ALGORITHMS

### 2.3.2 Dedmore algorithm

#### 1. Top corners (the X)

Start off by finding a starting corner. For example aim to solve the blue top face first. Rotate the cube to obtain the blue side of the corner in top position. Then move the blue center-piece into position without moving your corner. When this is done, rotate the whole cube to obtain the corner in your top left on the front side, as in the fig 2.5a.

Next you'll search for the next corner that should be positioned in the top right front side. Depending on the position on this corner, you will use different operation sequences to get it in position like the fig 2.5b.

Simply rotate the cube and continue on. The goal of this step is to form an 'X' on the top face as in fig 2.5c.

#### 2. Top edges

The goal of this step is to simply get the edges in place. Get them in place one by one. When this step is finished the cube looks like fig 2.6a.

#### 3. Middle layer

The next step is to solve the middle layer by moving down a cubie from the middle of the third layer to the correct position on the second layer (fig 2.3a).

#### 4. Bottom corners

Here the cube is turned upside down and the goal here is again to build a (green) X with the corners(fig 2.6b).

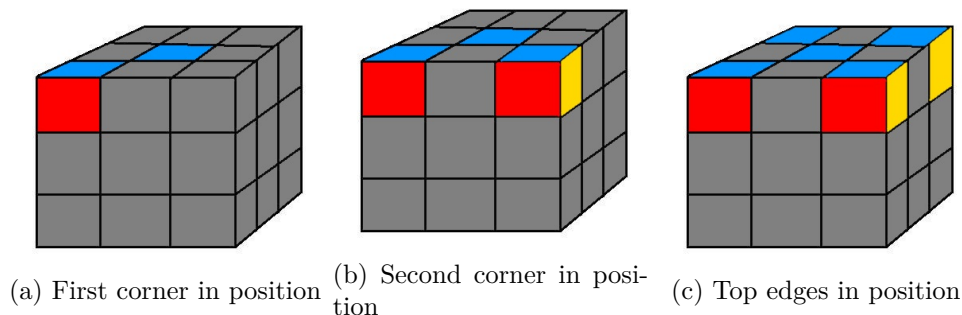


Figure 2.5

### 5. Bottom edges

In this step, at least one edge is already in the correct place (the color might be switched as in fig 2.6c). Find that edge and put it on the front side. Then position every other edge correctly by using a series of operations.

When every edge are correctly positioned, there are 2 possible states left. Each requires a different sequence of operations to solve. The ‘fish’ pattern (fig 2.7a) and the ‘H’ pattern (fig 2.7b). Execute the sequence for the pattern and you’ll then have solved the cube.

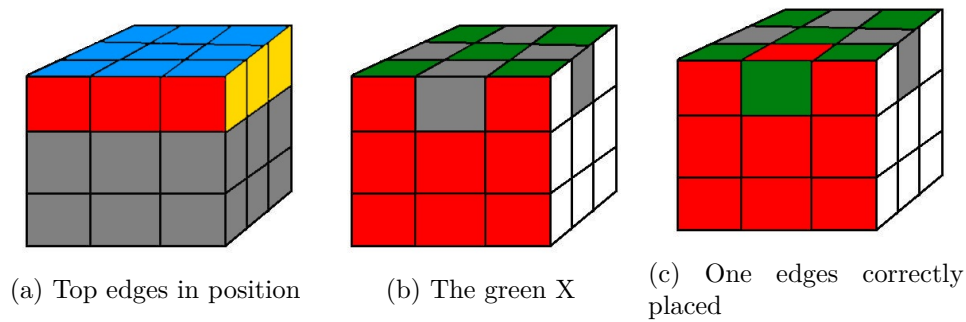


Figure 2.6

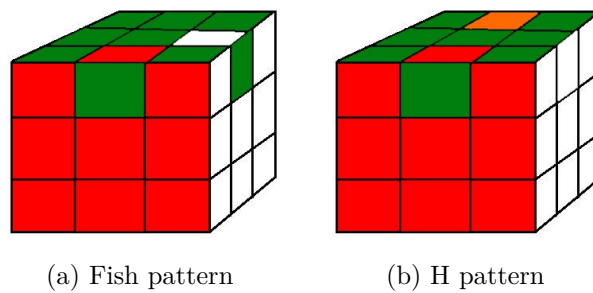


Figure 2.7

## **Chapter 3**

# **Method**

**3.1 Literature study**

**3.2 Implementation and data collection**

**3.3 Analyze and representation**





## **Chapter 4**

# **Implementation**

**4.1 Cube representation**

**4.2 Algorithms**

**4.3 Scramble**

**4.4 Difficulty**



## **Chapter 5**

# **Results and Analyze**

### **5.1 Data**

### **5.2 Comparison**



## **Chapter 6**

# **Discussion**

### **6.1 Comparison**

### **6.2 Errors**



## Chapter 7

# Conclusion

[6]





# References

- [1] Shelley Chang. How to solve the rubik's cube. <http://shellie.nfshost.com/cube/>, 2008. [Accessed 6 April 2015].
- [2] Herbert Kociemba. The two-phase-algorithm. <http://kociemba.org/twophase.htm>, 2014. [Accessed 6 April 2015].
- [3] kommer inte in i pp:n. Adde kolla på det. [http://faculty.mc3.edu/cvaughen/rubikscube/cube\\_counting.ppt](http://faculty.mc3.edu/cvaughen/rubikscube/cube_counting.ppt). [Accessed 6 April 2015].
- [4] Lars Petrus. Solving rubik's cube for speed. <http://lar5.com/cube/>, May 1997. [Accessed 3 April 2015].
- [5] Unknown. The history of the rubik's cube. <http://eu.rubiks.com/about/the-history-of-the-rubiks-cube>. [Accessed 6 April 2015].
- [6] Unknown. How products are made, volume 7 - rubik's cube. <http://www.madehow.com/Volume-7/Rubik-s-Cube.html>. [Accessed 3 April 2015].
- [7] WCA. Wca regulations article a: Speed cubing. <https://www.worldcubeassociation.org/regulations/#A3a1>, April 2014. [Accessed 3 April 2015].
- [8] WCA. Wca regulations article 12: Notation. <https://www.worldcubeassociation.org/regulations/#article-12-notation>, April 2014. [Accessed 3 April 2015].



## Appendix A

# Calculation of randomly rotate faces

Our calculations goes here

And here is a figure

Figure A.1: Several statements describing the same resource.

that we refer to here: A.1