Comparison of Rubik’s Cube Solving Methods Made for Humans

DANIEL DUBERG, JAKOB TIDESTRÖM
Comparison of Rubik’s Cube Solving Methods Made for Humans

Daniel Duberg
Jakob Tideström

Degree Project in Computer Science, DD143X
Royal Institute of Technology
Supervisor: Vahid Mosavat
Examiner: Örjan Ekeberg

Stockholm, Sweden 2015
Abstract

This study analyzed and compared four different methods of solving a Rubik’s Cube. Those four methods being the method on Rubik’s official website, the CFOP method, the Roux method and the ZZ method. The factors that were considered were the number of moves each method requires for solving a Rubik’s Cube, how many algorithms they require as well as how concrete or intuitive they are.

Our conclusion is that the CFOP, Roux, and ZZ method are fairly equivalent when it comes to move span, but CFOP has the lowest average number of moves used to solve a Rubik’s Cube. CFOP has more concrete algorithms and cases while both Roux and ZZ are more intuitive, ZZ uses fewer types of moves than Roux however.

The solution on Rubik’s official website does not compare, at its best it uses as many moves as the others do at their worst. It is however concrete and uses few algorithms for each part.
# Table of contents

1 Introduction ....................................................................................................................... 1  
1.1 Problem Definition .......................................................................................................... 1  
1.2 Problem Statement ......................................................................................................... 2  
1.3 Motivation ...................................................................................................................... 2  
1.4 Structure ....................................................................................................................... 2  
1.5 Terminology .................................................................................................................. 3  
2 Background ...................................................................................................................... 6  
2.1 Rubik’s Cube Solution Guide (RCSG) .............................................................................. 6  
  2.1.1 White Cross ............................................................................................................... 6  
  2.1.2 White Corners .......................................................................................................... 6  
  2.1.3 Middle Layer .......................................................................................................... 6  
  2.1.4 Top Layer .............................................................................................................. 7  
2.2 CFOP Method (Fridrich Method) .................................................................................... 9  
  2.2.1 The Cross ............................................................................................................... 9  
  2.2.2 F2L ...................................................................................................................... 9  
  2.2.3 Orienting of Last Layer .......................................................................................... 9  
  2.2.4 Permutation of Last Layer ...................................................................................... 9  
2.3 Roux Method .................................................................................................................. 10  
  2.3.1 First Block ........................................................................................................... 10  
  2.3.2 Second Block ....................................................................................................... 10  
  2.3.3 Remaining Corners ............................................................................................... 10  
  2.3.4 Remaining Edges .................................................................................................. 10  
2.4 ZZ Method ..................................................................................................................... 11  
  2.4.1 EOLINE ................................................................................................................ 11  
  2.4.2 F2L ...................................................................................................................... 11  
  2.4.3 Last Layer .......................................................................................................... 11  
3 Method .............................................................................................................................. 12  
3.1 Bidirectional Search ......................................................................................................... 12  
3.2 Implementation ............................................................................................................. 12  
3.3 Test Data Gathering Procedure ...................................................................................... 13  
3.4 Testing Environment ...................................................................................................... 13  
  3.4.1 Hardware ............................................................................................................... 13
3.4.2 Software ............................................................................................................. 13

4 Results ....................................................................................................................... 14

4.1 Number of Moves per Method ............................................................................... 14

4.1.1 Rubik’s Cube Solution Guide (RCSG) ............................................................... 14

4.1.2 CFOP Method ...................................................................................................... 14

4.1.3 Roux Method ...................................................................................................... 15

4.1.4 ZZ Method ......................................................................................................... 15

4.1.5 Compilation of All Methods .............................................................................. 16

4.2 Number of Moves per Part ................................................................................. 17

4.2.1 Rubik’s Cube Solution Guide (RCSG) ............................................................... 17

4.2.2 CFOP Method ...................................................................................................... 17

4.2.3 Roux Method ...................................................................................................... 18

4.2.4 ZZ Method ......................................................................................................... 18

5 Discussion ............................................................................................................... 19

5.1 Number of Moves per Method ............................................................................. 19

5.2 Number of Moves per Submethod ........................................................................ 19

5.4 Confounding Factors ............................................................................................ 20

6 Conclusion ............................................................................................................... 21
1 Introduction

Rubik’s Cube was created by Erno Rubik, a professor of architecture in Budapest, in 1974 as a means of teaching his students about spatial relationships. At the Nuremberg Toy Festival in 1979, the toy specialist Tom Kramer agreed to sell the Rubik’s Cube, then known as the Magic Cube, to the rest of the world [1]. It grew into the most popular toy in the world, with over 350 million units sold [2, p. 689]. Even though it is sold as a toy, it is more of a work of art to Rubik [1].

While the Rubik’s Cube comes in many variants, from 2x2x2 to 5x5x5 [3], the original version is the 3x3x3 cube [1]. The original version has $11! \times 8! \times 2^{12} \times 3^8$ (around 43 quadrillion) possible states [4] and is the version covered in this study.

In 2010 it took computers 35 CPU years to iterate through all of the possible states [5]. It was determined that God’s Number for a 3x3x3 Rubik’s Cube was 20 [2, p. 690].

IDA* search was used [5] which is an algorithm made for computers, not humans. Different methods have been suggested for solving a Rubik’s Cube that are feasible for a human to perform. Four of those methods have been observed, analyzed and compared in this study.

1.1 Problem Definition

The Rubik’s Cube is a complicated puzzle and one might wish to learn how to solve it in the most efficient way.

The most basic way of solving a Rubik’s cube is to use brute force, which is to test every possible combination of moves until the cube has been solved. There are other methods that aim to solve

---

1 Spatial: Relating to space [6]
2 Relationship: The way in which two or more people or things are connected [6]
3 CPU years: Work done by a 1 GFLOP reference machine in a year [7]
4 God’s Number: The minimum number of moves to solve any Rubik’s Cube [5]
the cube more efficiently. This study aims to investigate different aspects of a selection of those methods.

We could not find any previous work that covered this exact subject. The closest were about analyzing the methods themselves from a mathematical standpoint rather than comparing how well they solved a cube compared to each other.

There is also no scientific sources for the history or background for any of the methods covered, those who mention any of them use the same sources as in this study. This is the cause of the lack of academic studies as sources. The fact that when these methods are implemented they actually do solve scrambled Rubik’s Cubes is an indication of their validity.

1.2 Problem Statement

- How many moves\(^5\) on average does each method use to solve the cube?
- How many moves on average does each part of each method use to solve the cube?

1.3 Motivation

This study will be of interest to people who want to solve the Rubik’s Cube quickly and find the method that is best suited for them. It could also be of interest to those who want to see how different approaches to the same problem differ from each other.

1.4 Structure

The first section of this study introduces the reader to the subject with some background information about the Rubik’s Cube itself as well as the terminology that will be used throughout.

The second section will give some background to each of the chosen methods, the motivation behind their selection, and a description of each part of the solution.

The third section will describe the testing environment as well as the procedure used to gather the data.

The resulting test data will be tallied and presented in the fourth section and the final section will contain the analysis and discussion of said test data.

---

\(^5\) A move is defined in 1.5, Notation
1.5 Terminology

**Cell:** One of 54 colored squares that make up a Rubik’s cube. There are a total of nine cells per color.

**Face:** A cube has six faces where each face is made up of 3x3 cells. These faces are known as “Front”, “Top”, “Back”, “Down”, “Left” and “Right”.

**Piece:** One of the components of the Rubik’s cube itself. The 3x3x3 Rubik’s cube has 26 pieces.

**Corner:** Piece of the cube that has three cells on it. The cube has eight corners and thus eight corner pieces.

**Edge:** Piece of the cube that consists of two cells. There are 12 of these.

**Center:** The piece that holds the middle cell of each face, this is the piece that will determine what color a face is.

**State:** What colors each cell has across each face at a given point in time.

**Notation:**
The notation for permuting the cube that will be used in this report is described below in the table.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Image</th>
<th>Notation</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td><img src="image1.png" alt="Image" /></td>
<td>Li</td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td>R</td>
<td><img src="image3.png" alt="Image" /></td>
<td>Ri</td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
</tbody>
</table>

---

Component: A part of a larger whole, especially a part of a machine or vehicle. [6]
<table>
<thead>
<tr>
<th>U</th>
<th>Ui</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>Di</td>
</tr>
<tr>
<td>F</td>
<td>Fi</td>
</tr>
<tr>
<td>B</td>
<td>Bi</td>
</tr>
<tr>
<td>M</td>
<td>Mi</td>
</tr>
<tr>
<td>E</td>
<td>Ei</td>
</tr>
</tbody>
</table>
Table 1: Name and illustrations for each move notation.
2 Background

2.1 Rubik’s Cube Solution Guide (RCSG)
This method was selected because it is featured on the official Rubik’s Cubes website.

2.1.1 White Cross
Begin with the white center cell at the top face and create a white cross. There is no algorithm for this part [8, p. 3].

After the white cross is in place the next part is to make sure that the sides of the edge pieces match their respective faces center cell, without disturbing any of the other white cells in the cross. This is done by moving the edge piece down to the bottom layer, rotating the bottom layer until it is underneath the location where it should be, then move it up to the top layer [8, p. 3].

2.1.2 White Corners
The next part is to solve the white layer completely by orienting the corner pieces with a white cell on them. To do this the corner piece is placed on the bottom layer such that the corners other two colors matches the center cell of the adjacent faces. If the piece is already on the bottom layer it simply has to be rotated to the correct place. If however the piece is on the top layer it can be moved to the bottom layer with Ri, Di, R [8, p. 4].

Once the corner piece is in its intended location but on the bottom layer, twist the cube so that the corner is in the bottom right corner and repeat Ri, Di, R, D until the corner is correctly oriented. This entire process repeated for each of the 4 corners pieces [8, p. 4].

2.1.3 Middle Layer
The cube is now flipped so that the white layer that was previously solved is now on the bottom [8, p. 5].

To solve the middle layer, find an edge piece that does not consist of a yellow cell. If the piece is located at the top layer, move the top layer so that the cell of the piece on the side face is the same color as the center cell of the same side face [8, p. 5].

If the edge piece should be located to the right in middle layer, as seen in figure 3, then do U, R, Ui, Ri, Ui, Fi, U, F. If it belongs to the left, instead do Ui, Li, U, L, U, F, Ui, Fi, to move it to the right place in the middle layer. Do this until there are not any more edge pieces on the top layer that does not have any yellow cell [8, p. 5].

If one of the edge pieces in the middle layer is in the wrong position or orientation the wrong way then move it to top layer by doing the appropriate algorithm described above and repeat above [8, p. 5].

Figure 3: Illustration of state.
2.1.4 Top Layer

2.1.4.1 Yellow Cross
The next objective is to create a yellow cross on the top layer, colors on the side of the cube don’t matter for this part [8, p. 6].

If the only yellow cell on the top layer is the center cell then do \( F, U, R, U_i, R_i, F_i \) to reach either of the two states below [8, p. 6].

If there is a yellow horizontal line then do \( F, R, U, R_i, U_i, F_i \) [8, p. 6].

If the only yellow cells on the top layer is the top and left edge cell and center cell then do \( F, U, R, U_i, R_i, F_i \) again [8, p. 6].

2.1.4.2 All Yellow on Top
At this stage there are three states the cube can be in:

1. If no corner cells on the top layer are yellow then it should be oriented such that the bottom left corner piece, looking from above, has a yellow piece on the left face and then do the algorithm specified below [8, p. 7].

2. If only one corner cell is yellow, move that cell to the bottom left corner, again seen from above, and then do the algorithm below [8, p. 7].

3. The last possible state is that there are two yellow corner cells then one of the yellow cells on the sides should be positioned in the bottom left corner but with the yellow cell on the front face [8, p. 7].

Once the cube has been oriented as specified above, depending on the state, the algorithm \( R, U, R_i, U, R, U, U, R_i \) should be performed. After this algorithm has been performed, reorient the cube by repeating the instruction above and then do the algorithm again, this needs to be repeated until the entire top layer is yellow [8, p. 7].

2.1.4.3 Position the Yellow Corners
Start this part with putting at least two corners in the correct location by twisting the yellow layer. These correct corners must be placed at either A and B, A and D or B and C as seen in figure 4 [8, p. 8].

If the corners A and B are in the correct place but C and D need to be flipped then they can change places by doing the algorithm \( R_i, F, R_i, B, B, R, F_i, R_i, B, B, R, R, U_i \) [8, p. 8].

Figure 4: Positions of the yellow corners.
If the corners that need to be flipped is B and C or A and D then do the above algorithm once, put the correct corners in position A and B and then do the algorithm again [8, p. 8].

2.1.4.4 Position the Yellow Edges
The last part is to orient the yellow edge pieces correctly. If only one edge piece is correct, position this edge piece to the back face and determine if the remaining edge pieces need to be rotated clockwise or counter-clockwise. If no edge piece is correct then do either of the algorithms below and repeat this part [8, p. 9].

To rotate them clockwise use the following algorithm $F, F, U, L, Ri, F, F, Li, R, U, F, F$. If they need to be rotated counter-clockwise the algorithm is $F, F, Ui, L, Ri, F, F, Li, R, Ui, F, F$ [8, p. 9].
2.2 CFOP Method (Fridrich Method)
The name CFOP comes from the four stages used in the method, those being Cross, First Two Layers (F2L), Orient Last Layer (OLL), Permute Last Layer (PLL) [9].

This method is also known as the Fridrich Method since it was popularized by Jessica Fridrich [10]. This method was selected because it is well known and known as the most popular method for quickly solving a Rubik’s Cube.

2.2.1 The Cross
The first part of this algorithm is to solve the cross on the first layer, just like the RCSG method but here the cross does not have to be white. On average this takes about 7 moves to accomplish [10].

2.2.2 F2L
The next part is to orient the corner pieces on the first layer correctly while simultaneously orienting the edge pieces in the middle layer to correspond with the corner pieces. This also takes about 7 moves on average, for each of the corners [10].

2.2.3 Orienting of Last Layer
The second to last part is to orient the corner and edge pieces of the last layer simultaneous so that the last layer has the right color. There are 40 different algorithms for this part and takes about 9 moves to complete [10].

2.2.4 Permutation of Last Layer
The last part is to, without disturbing any of the corner or edge pieces, permute the remaining 8 pieces at once. There are 13 different algorithms for this part and it takes about 12 moves to complete [10].
2.3 Roux Method
This method is known as the Roux method after Gilles Roux who is credited as inventing it in 2003 [11].

The selection of this method was based on the fact that CFOP is reliant on algorithms while the Roux method is more intuitive. It is up to the solver to figure out how to do each part most efficiently, depending on the state, and not finding a specific case and executing a specific algorithm.

Since this is the case however the description of each part is short since there is no specific way to do it.

2.3.1 First Block
The first part is to select a face of the cube and solve the bottom two layers of the selected face. This will create a 1x2x3 block of solved cells on the cube [12].

2.3.2 Second Block
Next objective is to create another 1x2x3 block of solved cells, this time on the face opposite of the one selected in the first part [12].

2.3.3 Remaining Corners
On this part the remaining four corners on the top layer are simultaneously oriented and permuted. This will leave four edges unsolved on the top layer and two unsolved edges on the bottom layer which will be solved in the final part [12].

2.3.4 Remaining Edges
This part starts with orienting the remaining six edges correctly followed by solving the two unsolved edges on the top layer, this is to reduce the number of cases for the pieces left unsolved. As such, once those two edges have been solved, move on to solving the last four edges, there are only three possible cases for these pieces at this point [12].
2.4 ZZ Method
The ZZ Method is a modern speed solving method and was invented by Zbigniew Zborowski who published it in 2006 [13].

This method was selected because it is a more modern method than the previous two methods and it only uses three different moves for the majority of the solution.

2.4.1 EOLine
The ZZ Method starts off by orienting all of the edges, Edge Orientation (EO), while also placing the DF\(^7\) & DB\(^8\) edges, making a line. This is the most difficult part of the ZZ method and the author claims that it takes an average of 6 moves to perform [13].

2.4.2 F2L
The next part is to solve the bottom two layers of the faces that are adjacent to the line that was created in the previous part. This makes it possible to complete the last part of this method using only six moves, R, Ri, U, Ui, L, and Li [13].

2.4.3 Last Layer
There are different ways to solve the last layer, each with its own pros and cons [13]. COLL/EPLL is the method that was selected for this study because it has few cases relative to the number of moves required to solve it.

2.4.3.1 COLL
The goal of this part is to solve the corners of the last layer in one step. At this stage the cube can be in one of 40 possible states [14].

2.4.3.2 EPLL
The goal of the final part is to solve the remaining unsolved edges which can be done in one of four ways [14].

\(^7\) DF: Down-Front, the edge piece that has a cell on the front and down face.
\(^8\) DB: Down-Back, the edge piece that has a cell on the down and back face.
3 Method

Implementations of the chosen methods was not readily available, thus implementing them became part of the work. This gave greater knowledge and control of the implementations exact details as well as a way to save the sought after data.

3.1 Bidirectional Search

Some of the parts for each method require intuition and not well-defined algorithms. To find an optimal solution for these parts we decided to essentially brute force a solution by doing each move on an initial state and then trying each move again on the resulting states $n$ times until a solution was found [15, p. 36-72]. Since this results in $18^n$ states, where 18 is the number of possible moves from every state, the time required would grow exponentially.

When implemented as a breadth-first search [15, p. 36-41] memory ran out when a move sequence reached a length of around seven, which is often the case. An iterative deepening search [15, p. 65-72] was also attempted but if a sequence required eight moves then it would already take several minutes. Since this would have to be done multiple times per method it would have taken months or even years to get a sample size of over 10,000.

A bidirectional approach, where we apply the moves, as above, to both the initial state and the goal state until they reach a common state, to this problem allowed for twice the number of moves in the same time span [15, p. 73-74]. This means the number of states is $18^{n/2}$ for a move sequence of length $n$.

To further cut down on possible states we painted every irrelevant cell black so that the only pieces we needed to be part of the solution was considered, as seen in the pictures below.

![Figure 5: Example initial state.](image1)
![Figure 6: Example common State.](image2)
![Figure 7: Example goal state.](image3)

3.2 Implementation

Bidirectional search was implemented for the first two parts of each method, excluding RCSG where it was only implemented for the white cross.

Both CFOP's and ZZ's F2L had to be split into four subproblems. They both solve each corner-edge pair individually but ZZ also solves the bottom right and bottom left edge pieces at the same time.

Roux does the same for the second 1x2x3 block, it first creates a 1x2x2 in one of the corners and then the corresponding 1x1x2 block to finish the 1x2x3 block.
These were the more intuitive parts of each method, the rest had specific algorithms for solving specific cases and were implemented as described in [8], [10], [12], [13] and [14].

3.3 Test Data Gathering Procedure
To gather the test data that we need we perform 20-100 random moves on an already solved Rubik’s cube to get a random, but valid, shuffled cube for the method to solve. Each method is given an identical copy of this cube. Each method will be responsible for the gathering of the test data that is specific to them. We repeated this procedure 100,000 times to get a stable estimate of the average number of moves a method requires to solve any given cube.

3.4 Testing Environment
Each method solve the same cube so that they’re compared under equal circumstances, otherwise the data could potentially get skewed due to one method receiving easier or harder cubes to solve than others.

3.4.1 Hardware
CPU Intel Core i7 2630QM, 4 cores (8 logical processors), 2 GHz (2.8 GHz full power), 6 MB L3-cache, 1 MB L2-cache, 64-bit processor
GPU AMD Radeon HD 6770M (AMD official driver), 2 GB GDDR5
Memory 2 x 4096 MB DDR3 @ 1333 MHz

3.4.2 Software
Operating system: Windows 8.1 Pro 64-bit version 6.3.9600 build 9600
Java Version: 1.8.0_20
    Java(TM) SE Runtime Environment (build 1.8.0_20-b26)
    Java Hotspot(TM) 64-Bit Server VM (build 25.20-b23, mixed mode)
4 Results

4.1 Number of Moves per Method

4.1.1 Rubik’s Cube Solution Guide (RCSG)
Figure 8 shows the distribution of number of moves to solve the Rubik’s Cube with the RCSG method.

![RCSG - Number of moves](image)

Figure 8: How many times the RCSG method solves the Rubik’s Cube in $N$ moves.

4.1.2 CFOP Method
Figure 9 shows the distribution of number of moves to solve the Rubik’s Cube with the CFOP method.

![CFOP - Number of moves](image)

Figure 9: How many times the CFOP method solves the Rubik’s Cube in $N$ moves.
4.1.3 Roux Method

Figure 10 shows the distribution of number of moves to solve the Rubik’s Cube with the Roux method.

![Roux - Number of moves](image)

Figure 10: How many times the Roux method solves the Rubik’s Cube in \( N \) moves.

4.1.4 ZZ Method

Figure 11 shows the distribution of number of moves to solve the Rubik’s Cube with the ZZ method.

![ZZ - Number of moves](image)

Figure 11: How many times the ZZ method solves the Rubik’s Cube in \( N \) moves.
4.1.5 Compilation of All Methods
Table 2 shows a compilation of results, regarding number of moves, for all methods.

<table>
<thead>
<tr>
<th></th>
<th>Rubik</th>
<th>CFOP</th>
<th>Roux</th>
<th>ZZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>7</td>
<td>10</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>Maximum</td>
<td>227</td>
<td>70</td>
<td>81</td>
<td>79</td>
</tr>
<tr>
<td>Average</td>
<td>134.61</td>
<td>54.31</td>
<td>56.09</td>
<td>59.35</td>
</tr>
</tbody>
</table>

Table 2: Minimum, maximum, and average number of moves required to solve a Rubik’s Cube per method.

Figure 13 shows figure 9, 10 and 11 in the same figure for comparison, with a 99 percent confidence interval.

Figure 13: Comparison of figure 9, 10 and 11, excluding outlier cases.
4.2 Number of Moves per Part

4.2.1 Rubik’s Cube Solution Guide (RCSG)
Figure 14 shows how many moves the RCSG method used to solve each part of the method.

![Pie chart showing the number of moves for each part of the RCSG method.]

Figure 14: Number of moves used to solve each part using the RCSG method.

4.2.2 CFOP Method
Figure 15 shows how many moves the CFOP method used to solve each part of the method.

![Pie chart showing the number of moves for each part of the CFOP method.]

Figure 15: Number of moves used to solve each part using the CFOP method.
4.2.3 Roux Method
Figure 16 shows how many moves the Roux method used to solve each part of the method.

![Figure 16: Number of moves used to solve each part using the Roux method.](image)

4.2.4 ZZ Method
Figure 17 shows how many moves the ZZ method used to solve each part of the method.

![Figure 17: Number of moves used to solve each part using the ZZ method.](image)
5 Discussion
Since the intuitive parts of each method is implemented with a bidirectional search those parts will be solved optimally, with the exception of the parts that had to be split into subproblems. The other parts were implemented exactly as specified which means that once a person becomes proficient in the use of a method, these are the results they can come to expect.

A human can potentially get better results by merging parts together, for example solving one or more of the corners as they are constructing the cross. This factor will differ between people and states so on average a person should expect results along the lines of the ones we have obtained.

5.1 Number of Moves per Method
The method that uses the highest number of moves is the RCSG method that solved a Rubik’s Cube with 227 moves. Barring the outlier cases, the RCSG method solved a Rubik’s Cube in 59 moves at best, which is around what the other methods used on average.

This is understandable since the other methods are methods made for experts while the RCSG method is made for beginners and thus favors the easily explained and understood algorithms over the optimal, and more complex, algorithms.

While the Roux method managed to solve a Rubik’s Cube with eight moves, and CFOP with ten, and ZZ with five, these were outlier cases and otherwise all three expert methods solved the Rubik’s Cubes in about the same interval.

5.2 Number of Moves per Submethod
The part with the least amount of moves for each method is the first part. This is because no pieces of the Rubik’s Cube have been solved yet. This lets the algorithm disregard the state of the other pieces and just find the shortest path to its goal.

The white corners and middle layer part of the RCSG method is equivalent to the F2L parts of the ZZ and CFOP method. Nearly half of the moves for each of these methods go to solving F2L. This is because F2L represents solving 61 percent of the Rubik’s Cubes cells.

Even though the first two parts of the Roux method has the same objective, the second block takes almost twice the moves to solve. This is because while the objective is the same, the second block has the added constraints of not being allowed to impact the cells that construct the first block.

Both the RCSG method and the CFOP method start out with solving the cross. The reason why CFOP does this quicker than the RCSG method is because the RCSG method specifies that it is the white cross that is to be solved while the CFOP method simply wants a cross of any color.
5.4 Confounding Factors

The most significant confounding factor is the fact that not every part of every method could be
done using the bidirectional search method we outlined earlier. This leads to the algorithm
potentially having to waste moves to make the Rubik’s Cube match any of the pre-defined states
and then executing the algorithm from there instead of simply immediately taking the shortest
path.

Related to this is the fact that some parts could be solved by the bidirectional search but had to
be split up into smaller subproblems first. This means that while it will find the shortest path for
each of these subproblems, it could possibly have solved the problem as a whole in fewer moves.
An example of this is F2L in the CFOP method, we divided the problem into four edge and
corner pairs. It would check which pair it could solve with the fewest number of moves, solve it
and then repeat this for the remaining corners. Solving one corner could negatively impact the
solution for any of the other corners. That would not have been an issue with a bidirectional
search that solved all pairs at once.

There is nothing stopping the shuffling process from negating the previous move by doing its
exact opposite, this could lead to outlier states where the cube is barely shuffled at all. It could
also give one method an unfair advantage over the others. An example of this is the state where
the Roux method managed to solve the whole cube in only eight moves, after the first block was
made it only had to perform one more move to solve the cube completely.
6 Conclusion

The immediate conclusion to draw is that the RCSG method is not comparable to the other three methods. This is understandable since it is intended for novices whose knowledge is not necessarily at the point required for the other more complex methods.

Barring the outlying case mentioned above, each of these three methods have approximately the same lower bound when it comes to solving a Rubik’s Cube but the ZZ method has a higher upper bound than the CFOP method and the Roux method has an even higher upper bound than the ZZ method. This gives the CFOP method the lowest number of moves required to solve a Rubik’s Cube on average.

Another thing to consider is that CFOP is more a concrete method with specific cases and specific algorithms while ZZ and Roux are more intuitive. The difference between ZZ and Roux is that ZZ only uses some of the available moves which require the consideration of fewer move sequences.
References


