

2013-04-12  
Stockholm

# Benchmarking Rubik's Revenge algorithms

Paolo Brolin Echeverria - [paolobr@kth.se](mailto:paolobr@kth.se) - 0735 44 10 70

Joakim Westermark - [jwestern@kth.se](mailto:jwestern@kth.se) - 0762 06 61 50

KTH, CSC - DD143X

Degree Project in Computer Science, First Level

Supervisor: Johan Boye

## **Abstract**

This Bachelor thesis paper investigates 2 different methods used to solve the Rubik's Cube 4x4x4 puzzle. The analyzed methods are Reduction and Big Cube method. We have implemented the cube and the two solvers in Python. Through a series of tests we have concluded that the Big Cube method has a better average move count as well as a low standard deviation in comparison to the Reduction method. However the reduction method has a lower minimum move count and consists of fewer algorithms. The best approach would be to combine both methods to form an optimal solution.

## **Sammanfattning**

Denna kandidatexamensuppsats undersöker två olika metoder som används för att lösa Rubiks Kub 4x4x4. Metoderna som analyseras är Reduction och Big Cube. Vi har implementerat kuben samt de bägge lösarna i Python. Genom en serie tester har vi kommit fram till att Big Cube har ett lägre genomsnittligt rotationsantal samt lägre standardavvikelse än Reduction. Reduction-metoden har däremot ett lägre minimumvärde på antalet rotationer och består av färre algoritmer. Det bästa tillvägagångssättet vore att kombinera de båda lösningarna.

## Table of Contents

<b>Abstract</b> .....	<b>i</b>
<b>Sammanfattning</b> .....	<b>i</b>
<b>Introduction</b> .....	<b>1</b>
<b>Context</b> .....	<b>1</b>
<b>Purpose</b> .....	<b>1</b>
<b>Background</b> .....	<b>2</b>
<b>The cube</b> .....	<b>2</b>
Structure .....	2
Complexity .....	3
Notation.....	4
Competitions.....	6
Scrambling .....	6
<b>Solving methods</b> .....	<b>6</b>
Reduction.....	6
Big Cube Solution .....	12
Parities.....	15
<b>Method</b> .....	<b>17</b>
<b>Implementation</b> .....	<b>17</b>
The Cube .....	17
Algorithms.....	17
<b>Benchmark tests</b> .....	<b>17</b>
Test design .....	17
<b>Results</b> .....	<b>18</b>
<b>Big Cube</b> .....	<b>18</b>
<b>Reduction</b> .....	<b>19</b>
<b>Discussion</b> .....	<b>25</b>
<b>Move expectancy and variation</b> .....	<b>25</b>
<b>Parity</b> .....	<b>25</b>
<b>Complexity of the methods</b> .....	<b>26</b>
<b>Sources of error</b> .....	<b>26</b>
<b>Conclusions</b> .....	<b>26</b>
<b>Appendix A</b> .....	<b>27</b>
<b>Appendix B</b> .....	<b>28</b>
<b>Appendix C</b> .....	<b>29</b>

# Introduction

## Context

This paper is our Bachelor Thesis report for Computer Science and Engineering programme at the School of Computer Science and Communication, KTH Royal Institute of Technology.

We will research a variant of the famous puzzle Rubik's Cube and methods commonly used to solve the puzzle. Rubik's Cube is a 3-dimensional combination puzzle and was invented 1974 by Hungarian sculptor and professor of architecture Ernő Rubik.<sup>1</sup> The puzzle has since been the world's most sold puzzle game with more than 350 million cubes sold worldwide.<sup>2</sup> Organized competitions and tournaments in speed-solving the puzzle has been held since the beginning of the 80's. Since 2004 the competitions has been organized by the World Cube Association which provides the full set of regulations and guidelines used in the Rubik's Cube community.<sup>3</sup>

In the Rubik's Cube community a sequences of moves in order to come closer to a solution of the puzzle is commonly called an Algorithm. Similar to the mathematical definition of an algorithm it denotes a well defined list of instructions to complete a task, given a initial state - through well defined stats - to a desired end state.<sup>4</sup>

## Purpose

The purpose of this paper is to describe and test methods that that solves the Rubik's Revenge puzzle. We focus on methods that are commonly used by top performing speed-solving practitioners. Based on test data we will evaluate each method's rotation variance, expected number of rotations and the frequency of parity errors in order to find the method that has the least expected number of moves. Depending on the expected number of rotations a method with 100-200 moves per solution might be better or worse than an method that uses 80-300 moves. The latter will surely result in more world records than the 100-200 method, but which one will for example win the most competitions?

---

1 John Nadler, "Squaring Up to the Rubik's Cube", <http://www.time.com/time/arts/article/0,8599,1669535,00.html>, Time, October 09, 2007

2 William Lee Adams (2009-01-28). "The Rubik's Cube: A Puzzling Success", <http://www.time.com/time/magazine/article/0,9171,1874509,00.html>, TIME. Retrieved 2009-02-05.

3 World Cube Association, <http://www.worldcubeassociation.org/>, 2008-04-09. Retrieved 2008-04-16

4 Stone, Harold S. (1972). Introduction to Computer Organization and Data Structures (1972 ed.). McGraw-Hill, New York. ISBN 0-07-061726-0

# Background

## The cube

### Structure

There are numerous variations of the Rubik's cube, stretching from 2x2x2 cubes up to 7x7x7 and beyond. The only limit of the cube is its axis, meaning the cube could potentially exist in all  $n \times n \times n$  forms where  $n$  is a positive integer. Other Rubik's Cube based spin-offs of different sizes and shapes also exists, such as the Pyraminx (pyramid shape) and many others.<sup>5</sup>

The original Rubik's Cube models a 27-piece puzzle with 54 (9 stickers on each of the 6 sides) colored stickers in the shape of a cube. However the structure of the Rubik's Cube is a bit different from what it is portraying. The Rubik's cube consists of 26 pieces (there is no middle piece) and there are 3 different types of pieces: centerpieces, edgepieces and cornerpieces.<sup>6</sup>

This report concerns the 4x4x4 Rubik's cube variation (Rubik's revenge). There are a few differences between the 3x3x3 cube and the 4x4x4 cube, the obvious difference being the puzzles they are modeling. The original Rubik's cube is a 27 (26 in reality) piece cube with 54 colored stickers and the Rubik's revenge is a 64 (56 in reality) piece cube with 96 colored stickers.<sup>7</sup>

As briefly mentioned in above the 3x3x3 cube consists of 3 different types of pieces. The 4x4x4 cube also consists of 3 different types of pieces. These pieces can be seen in figure 1.

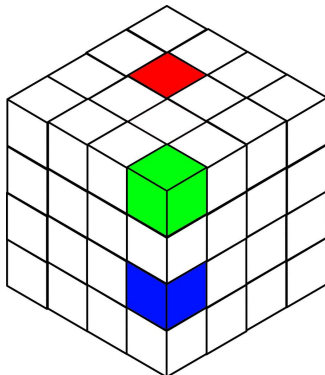


Figure 1 - Pieces

---

5 Dou, Daniel. 62 unique Rubik's Cubes, <http://www.theendearingdesigner.com/62-unique-rubiks-cubes>. Retrieved 2013-04-12

6 Rubik, Ernő. Patent filing, no. 289192, 1983-03-29.

<http://www.cs.brandeis.edu/~storer/JimPuzzles/RUBIK/Rubik3x3x3/PATENTS/Rubik4378116.pdf>

7 Sebesteny, Peter. Patent filing, no. 343732, 1983-12-20. <http://www.freepatentsonline.com/4421311.pdf>

### *Centerpieces (red in figure 1)*

There are 24 centerpieces allocated in 24 center positions. Any given centerpiece can be rotated into any center position (there are a few permutation sets that cannot happen).

### *Edgepieces (blue in figure 1)*

There are 24 edge pieces allocated in 24 edge positions. Like the centerpieces the edge pieces can be rotated into any edge position (again, there are a few permutation sets that are not possible).

### *Cornerpieces (green in figure 1)*

There are 8 cornerpieces allocated in 8 corner positions. They can also rotate into any given corner position with a few exceptions.

The different pieces cannot rotate into each others positions. E.g. a cornerpiece can never reside in a center position. They are simply not the same kind of piece, a centerpiece has only one sticker, an edgepiece has two stickers and a corner piece has 3 stickers. See figure 1.

All colors also have fixed positions in a solved cube. Blue can for example never end up next to green, as they are opposite colors. Yellow and white are also opposite colors, as well as red and orange.

## **Complexity**

Why do we use algorithms to solve the cube? While it doesn't seem like it at first glance, the Rubik's puzzle is very complex.

### *Permutations*

The complexity of the cube problem is vast. For example the number of permuted states (different states that can be reached through unique scrambles) that a 3x3x3 cube can reach is  $4.3 \cdot 10^{19}$ .<sup>8</sup> For the 4x4x4 cube the number is  $7.4 \cdot 10^{45}$ .<sup>9</sup> As can be seen the complexity greatly rises with only one extra piece (per side) in the puzzle. Moreover, from every single state there are 18 different moves that can be made (not counting flipping the cube over on its side).

### *God's number*

God's number is a popular name for the maximum number of moves needed to solve a cube from any given scramble ("could God do better"). God's number for the 3x3x3 cube is 20 moves.<sup>10</sup> This number was discovered in 2010 by doing an exhaustive search of all permutations (all possible scrambles) of the cube. Then all move-sets up to 20 moves were tested so that they could see that all scrambles indeed had a solution within 20 moves. With the

---

8 Counting the Permutations of the Rubik's Cube, Scott Vaughen. Professor of Mathematics. Miami Dade College.

[http://faculty.mc3.edu/cvaughen/rubikscube/cube\\_counting.ppt](http://faculty.mc3.edu/cvaughen/rubikscube/cube_counting.ppt). Retrieved 2013-04-12

9 Scherphuis, Jaap. Rubik's Revenge / Master Cube / 4x4x4 Rubik's cube, <http://www.jaapsch.net/puzzles/cube4.htm>. Retrieved 2013-04-12

10 Rokicki, Kociemba, Davidson and Dethridge. God's number is 20, <http://www.cube20.org/>. Retrieved 2013-04-12

current technology and processing power this is not practically possible to do on a 4x4x4 cube. The numbers of permuted states are far too many.

### *Algorithms*

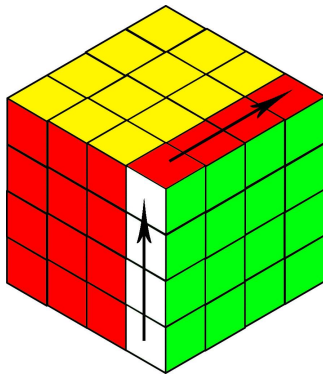
The complexity of the algorithms within the methods used for solving the cube is also greater on the 4x4x4 version (compared to the 3x3x3 cube). Algorithms systematically solve the cube step by step. The number of moves necessary to solve the cube using algorithms exceeds Gods number by many. But algorithms are basically heuristics guaranteeing a solution. Using algorithms a human can solve the cube within minutes while an exhaustive search by a computer could take much longer time to find a solution. These algorithms are shown as sequences, e.g. “R F’ L Rw r’ F2” together with a picture that shows what the algorithm accomplishes.

### **Notation**

As stated above, algorithms are described as sequences of letters and numbers. These letters and numbers can differ between different algorithm notations. We have been using the standard WCA notation.<sup>11</sup> The notation is described as follows:

#### *Outer axis*

R = clockwise 90 degree rotation of the far right axis of the cube (when holding it in front of you). This goes for all single uppercase letters.



**Figure 2 - Right outer axis rotation**

- R = Right
- L = Left
- F = Front
- B = Back
- U = Up
- D = Down

---

<sup>11</sup> The World Cube Association. Regulations – Notation, <http://worldcubeassociation.org/regulations/#article-12-notation>. Retrieved 2013-04-12

A single uppercase letter followed by a "2" means the rotation should make a 180 degree turn instead of a 90 degree turn. E.g. "L2", a 180 degree clockwise far left axis turn.

*Inner axis*

A single lowercase letter means the rotation should occur on the inner letter axis. E.g. "r", a 90 degree clockwise inner axis turn (without moving the outer axis).

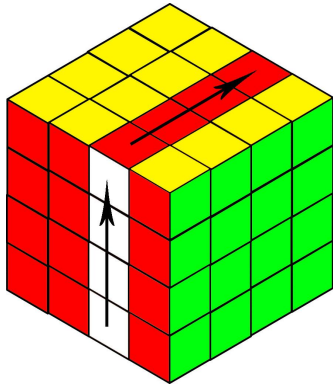


Figure 3 - Right inner axis rotation

*Slice*

A single uppercase letter followed by a "w" means the rotation should include both axis of the letter. E.g. "Rw", a 90-degree clockwise far right axis turn and an inner right axis turn, this is called a slice (we slice the cube in half when rotating two sides as once).

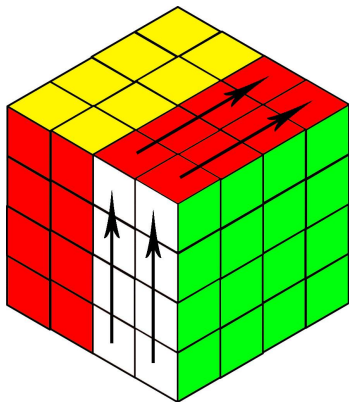


Figure 4 - Right slice rotation

Any letter followed by a prime ' is to be rotated anti-clockwise. E.g. R', a 90 degree anti-clockwise far right axis turn.



## Competitions

The Rubik's cube has a fairly large competitive scene as the puzzle has excellent competitive attributes.<sup>12</sup> The cube is a complex puzzle where dexterity is key in both hand and mind, making it interesting to measure results in a competitive way. Using a wide arrangement of methods with many different algorithms and techniques competitors can test their skills in speed solving, the art of solving a cube as fast as possible.

The World Cube Association (WCA) governs competitions for all puzzles labeled as Rubik puzzles, and all other puzzles that are played by twisting the sides. Selections of these puzzles are chosen as official events of WCA. The World Cube Association organizes competitions all over the world, supported by national organizations responsible for organizing competitions in their countries.<sup>13</sup>

## Scrambling

Competition scramble sequences must be generated using the current official version of the official WCA scramble program, which is available via the WCA website.<sup>14</sup>

## Solving methods

We will be comparing two different sorts of methods that both solve the 4x4x4 cube puzzle. There are many different methods and an immeasurable amount of variations to these. The two methods that are used the most in competitions when looking at record holders and the best performing competitors during competitions are called Kirjava and the Yau method. However, it is very difficult to say exactly which variation (their own or adapted) that they are using. We will be looking at two generic versions of these and how they differ in their way of solving the puzzle. We will also be looking at what the differences are in terms of moves expectancy and parity frequency.

The Yau method is derived from a method called Reduction and the Kirjava method is derived from a method called Big Cube. These two methods are explained below.

### Reduction

This method reduces the 4x4x4 cube to a 3x3x3 cube, which is much easier to solve. The way it works is as follows:

#### *Step 1.*

First we complete 4 centerpieces and their opposite 4 centerpieces. This is done intuitively and therefore there are no standard algorithms for this.

---

12 The World Cube Association. Competitions, <http://www.worldcubeassociation.org/results/competitions.php>. Retrieved 2013-04-12

13 The World Cube Association. Mission, <http://www.worldcubeassociation.org/mission>. Retrieved 2013-04-12

14 The World Cube Association. Scrambles, <http://www.worldcubeassociation.org/regulations/scrambles>. Retrieved 2013-04-12

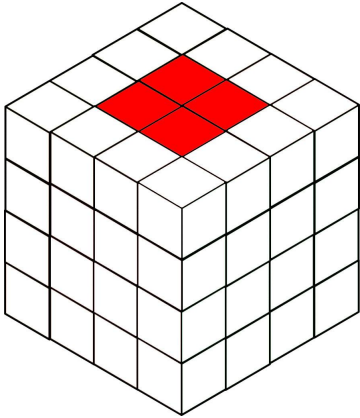


Figure 5 - Side up

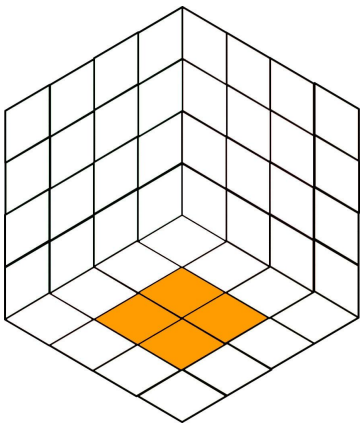


Figure 6 - Side down

These two cubes are two sides of the same cube. In step one only the two opposite centers are solved i.e. orange and red, blue and green or white and yellow.

*Step 2.*

In the next phase we complete all the other centerpieces. This is also done intuitively.

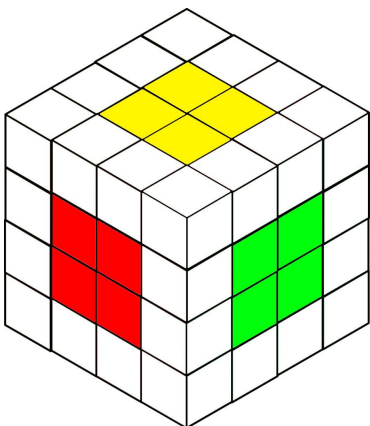
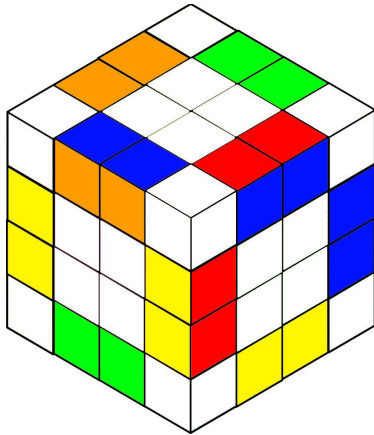


Figure 7 - Centerpieces

Now when all centerpieces are complete, these pieces have been reduced from 6\*4 centerpieces to 6\*1 centerpieces.

*Step 3.*

Now we pair up all edgepieces with their corresponding neighbor. This can be done intuitively as well but there are available algorithms to be used for this step.



**Figure 8 - Edgepieces**

*All edgepieces are paired up, but not oriented correctly*

When all edgepieces are paired up, the cube has successfully been reduced to a 3x3x3 cube. The edgepairs all count as one piece instead of two pieces, the centers all count as one centerpiece instead of four and the corners are the same.

**3x3x3**

From now on we only rotate the outer axes of the cube (i.e. no slices). By doing this the edgepairs and centerpieces stay intact. All steps from now on are the same as for a 3x3x3 cube. We have chosen the most widely used method here as well. It is called the Fridrich method.<sup>15</sup> The 3x3x3 cube solution is a layered solution, meaning we complete the cube layer by layer, see figure 9.

---

<sup>15</sup> Fridrich, Jessica. My system for solving Rubik's cube, <http://ws2.binghamton.edu/fridrich/system.html>. Retrieved 2013-04-12

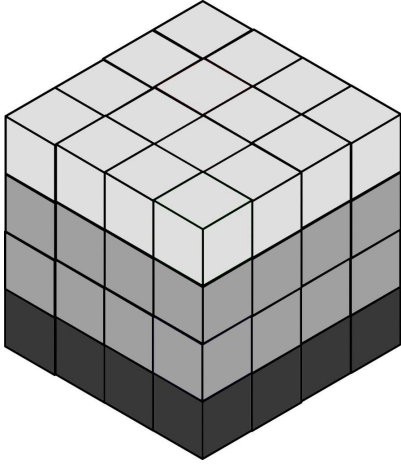


Figure 9 - 3 layers

*Step 4.*

First we make a cross on one side. The centers and edgepieces were already completed in step 2 and 3. Step 4 only puts 4 edges into position.

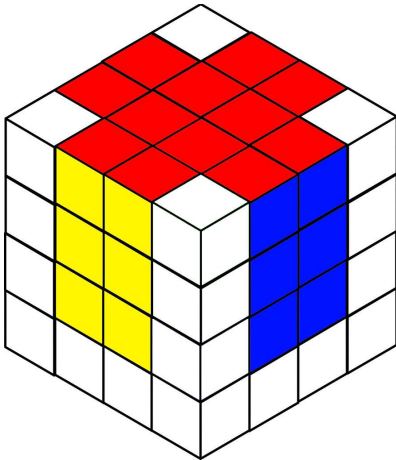


Figure 10 - Cross

*Step 5.*

The next step will complete the first layer and the second layer at once. This almost cuts the number of moves in half compared to solving the layers separately.

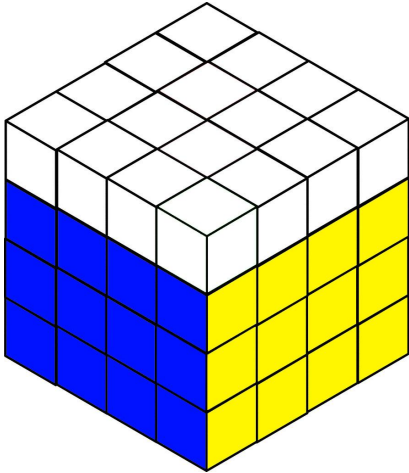


Figure 11 - First 2 layers

*Step 6.*

Now that the first two layers are complete, we complete the last layer in four steps.

**Step 6.1**

We make a cross on the top layer.

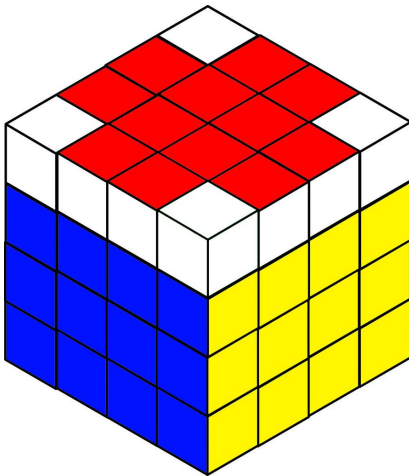


Figure 12 - Last Layer cross

**Step 6.2**

We rotate all the corners so that the right color is up. The entire top side is complete, but the edges and corners of the top layer is yet to be positioned right.

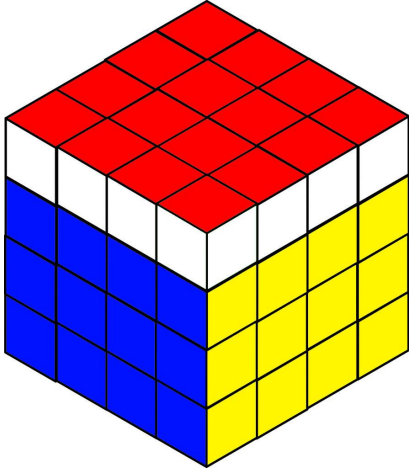


Figure 13 - Last Layer up complete

**Step 6.3**

We permute all the corners so that they reside in their right slots.

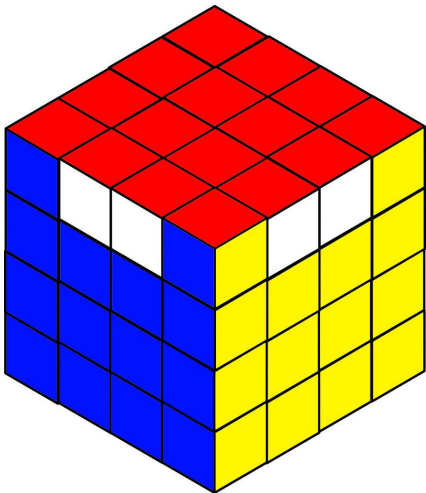


Figure 14 - Last layer corners complete

**Step 6.4**

We move all the edgepieces in the top layer to their respective position. This step can in many cases be completed in step 3.

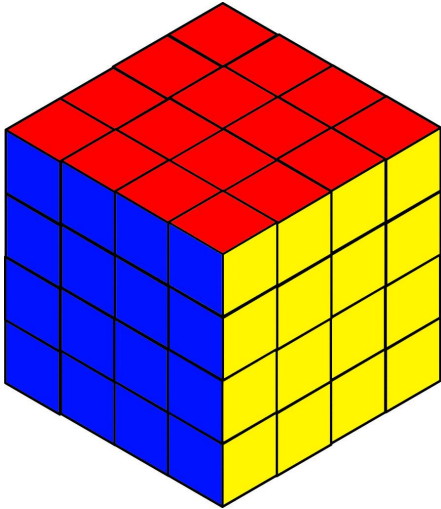


Figure 15 - Cube solved

### Big Cube Solution

This solution also reduces the puzzle from 4x4x4 to a 3x3x3 puzzle, but not all the way. This solution doesn't reduce the edges by pairing them up like in the reduction method. As can be seen later in step 2, only the edges for one color are paired up.

#### Step 1.

The first step is just like the one for reduction. We complete the opposite centers followed by the remaining centers. See step 1 and 2 in reduction.

#### Step 2.

In the next step we pair up edges just like in reduction but only for one color.

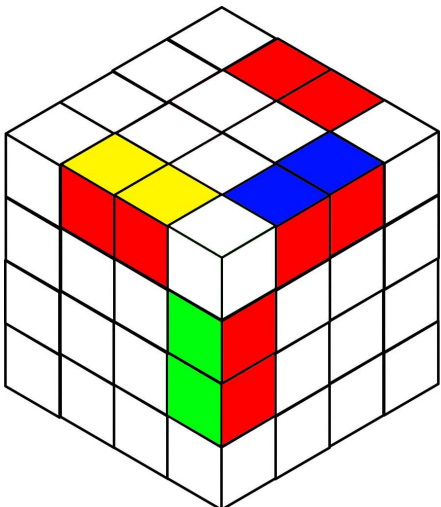


Figure 16 - One color edges

#### Step 3.

Now that these 4 edgepieces are paired up we complete a cross using them. This is just like in step 4 for the reduction method.

*Step 4.*

What we have now is an almost complete first layer. We now insert the first layers corresponding corners.

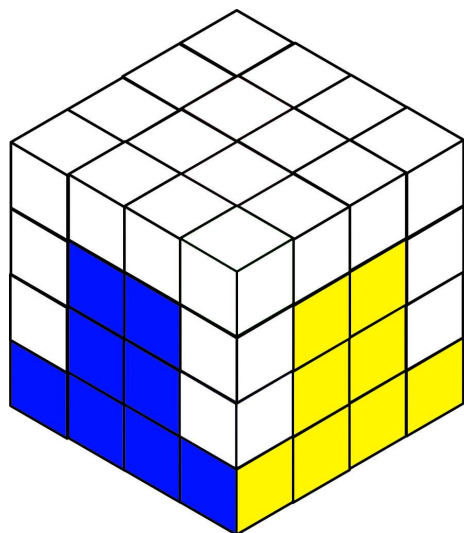


Figure 17 - First layer

*Step 5.*

First layer is done and we move on to the second and third layer (in this solution we have four layers since we didn't reduce the cube to a 3x3x3 cube). We insert all edgepieces one by one and sometimes 2 at a time, utilizing symmetries.

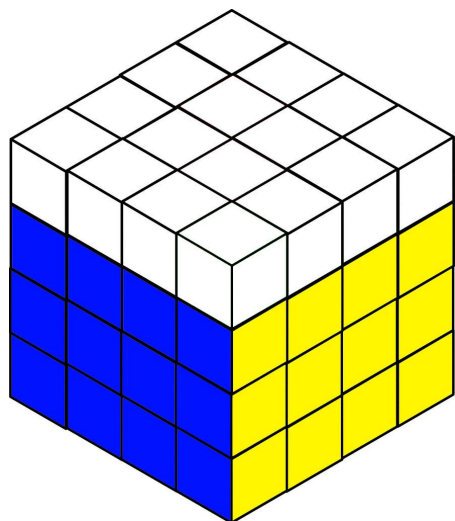


Figure 18 - First 3 layers



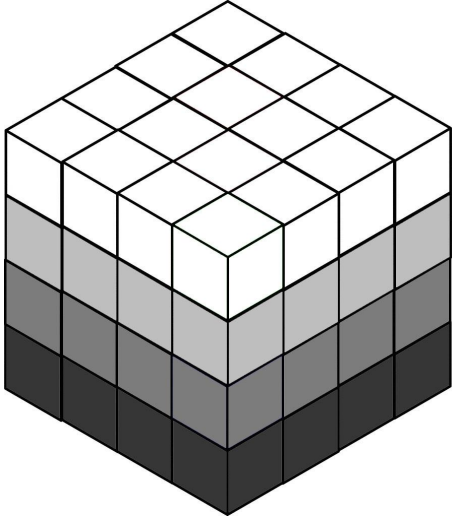


Figure 19 - 4 layers

*Step 6.*

As pictured in figure 18 we now only have the last layer left. The difference from the reduction method is that the edgepieces are not paired up, unless we get extremely lucky. We start out by turning the corners so that they have their right color facing upwards. The red centerpiece was completed in step 1.

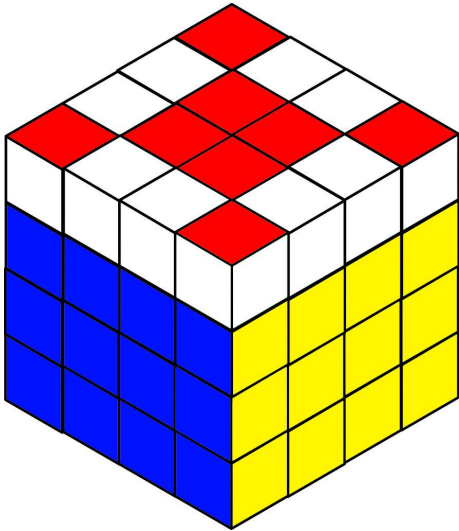


Figure 20 - Last layer corner turn

*Step 7.*

In this step we put the right corners in their right positions.

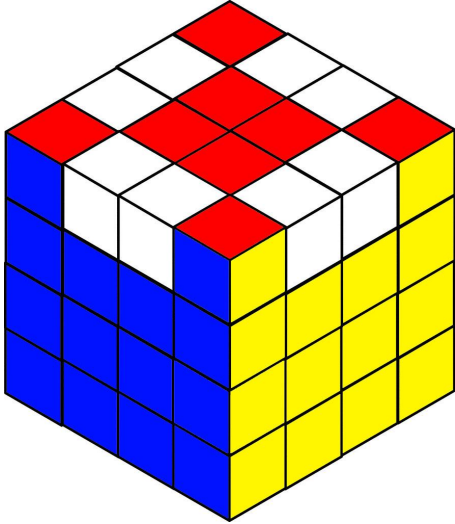


Figure 21 - Last layer corners

*Step 8.*

Next step is to insert the remaining edgepieces in their correct positions and turning them right in the process.

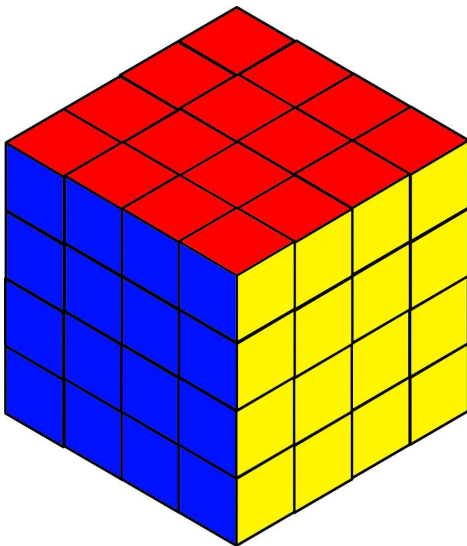


Figure 22 - Cube is complete

*Cube is solved.*

### Parities

Using only the steps described above could result in a failed solution caused by parity error. Parity is a misnomer cubing term that is used colloquially to describe an odd permutation of pieces within a certain defined orbital. It is most commonly used to refer to certain situations that can arise on the 4x4x4 cubes when solving with the reduction method. If it were the original 3x3x3 cube we were solving, and not a reduction of the 4x4x4 cube, the parity wouldn't exist. The Big cube solution can just like the reduction method be subjected to a parity error. There

are two different types of parity errors: Permuting Last Layer Parity (PLL) and Orienting Last Layer Parity (OLL). Both parity errors can be corrected with algorithms, resulting in a high move count.

### *PLL Parity*

PLL parity error is when two edgepairs have the wrong positions (but the cube is in all other regards solved).

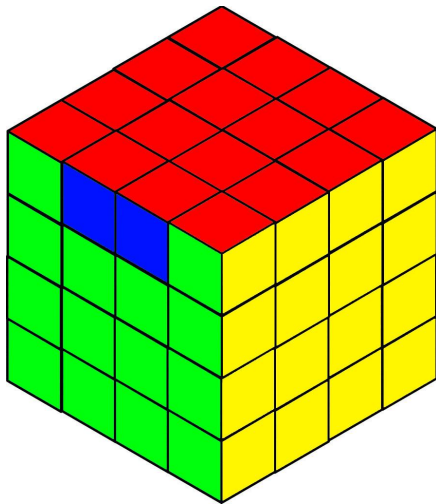


Figure 23 - PLL parity front side

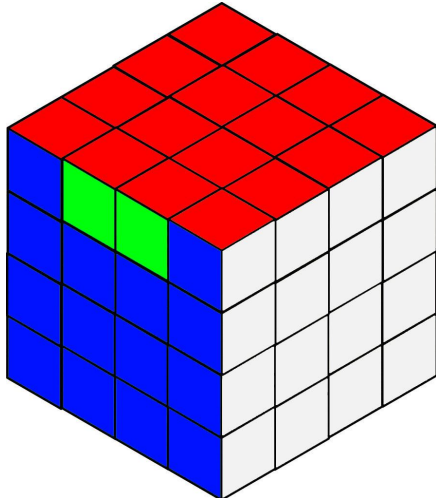


Figure 24 - PLL parity back side

### *OLL Parity*

OLL parity error is when one edgepair is in its right position but flipped.

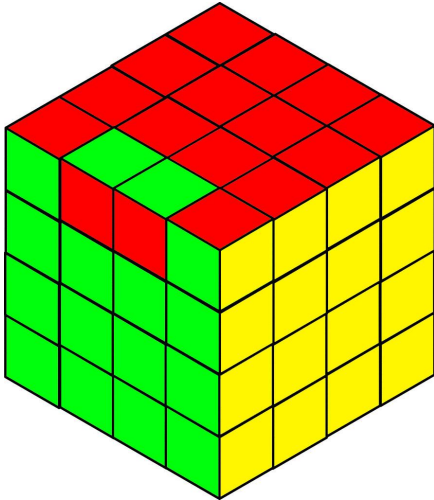


Figure 25 - OLL parity

## Method

### Implementation

#### The Cube

We have modeled a 4x4x4 cube in python programming language. The cube can rotate and spin just like a physical cube with the same limitations. The cube takes input on the notation form described in the notation section.

#### Algorithms

We have implemented the two solvers, which solve the cube in steps accordingly with the steps in the Solving methods section. Each solver evaluates which step it is on (from the Solving methods section) and looks for the appropriate algorithm to execute. When done with each step the solver reevaluates the cube and looks for the next appropriate algorithm to use. This chain runs until the cube is solved.

### Benchmark tests

#### Test design

We have downloaded 10 000 unique scrambles from the World Cube Association scrambler. Due to data corruption 4 scrambles were rendered unusable. Each solver then solved each of the scrambles generating test data.

In 9996 scrambles for each method we have measured the following:

- The total number of moves used for solving the cube.

- Number of moves in each sub-step of the solutions (the sub-steps can consist of one or several algorithms).
- Parity error occurrence.

The first two steps in the solvers are exactly the same - completing the opposite centers and then completing the rest of the centers - so these were not measured. The measuring starts were the solvers differ.

The tests were run on a standard desktop computer and took approximately 1 hour to run. The computer ran windows 7 and python 3.3.0. The hardware is irrelevant since we did not measure the time it took the computer to solve the cubes. We only measured the number of moves it took. The runtime could have been 10 seconds on a supercomputer or 1 day on an old desktop computer, the number of moves would still be the same. We ran the same test 3 times and the results were as expected the same on all runs.

## Results

The solvers successfully solved 9996 cubes with the following results.

### Big Cube

Table 1 - Big Cube test data

	<i>Average number of moves</i>	<i>Max moves</i>	<i>Min Moves</i>	<i>Standard Deviation</i>
<b>Total</b>	160.6310155	199	110	11.72441878
<b>Cross</b>	7.953776888	17	0	2.281444166
<b>Edges</b>	27.34037019	35	5	3.815014897
<b>Layer 1</b>	16.88194097	28	3	4.111010737
<b>Layers 2 &amp; 3</b>	58.72546273	64	24	6.312921571
<b>Last layer</b>	49.72946473	75	12	7.675049909

Table 2 - Big Cube Parity data

	<b>Total</b>	<b>Prob</b>
<b>Paritys</b>	2643	26%
<b>Single parity</b>	1997	
<b>Double parity</b>	323	
<b>PLL</b>	813	8%
<b>OLL</b>	1830	18%

## Reduction

Table 3 - Reduction test data

	<b>Average number of moves</b>	<b>Max moves</b>	<b>Min Moves</b>	<b>Standard Deviation</b>
<b>Total</b>	166.4381191	247	96	20.8945734
<b>Cross</b>	8.913856928	17	0	2.70519388
<b>Edges</b>	70.97358679	99	29	11.06408383
<b>Layer 1 &amp; 2</b>	36.90505253	66	14	7.743905367
<b>Last Layer</b>	49.64562281	95	8	15.69034709

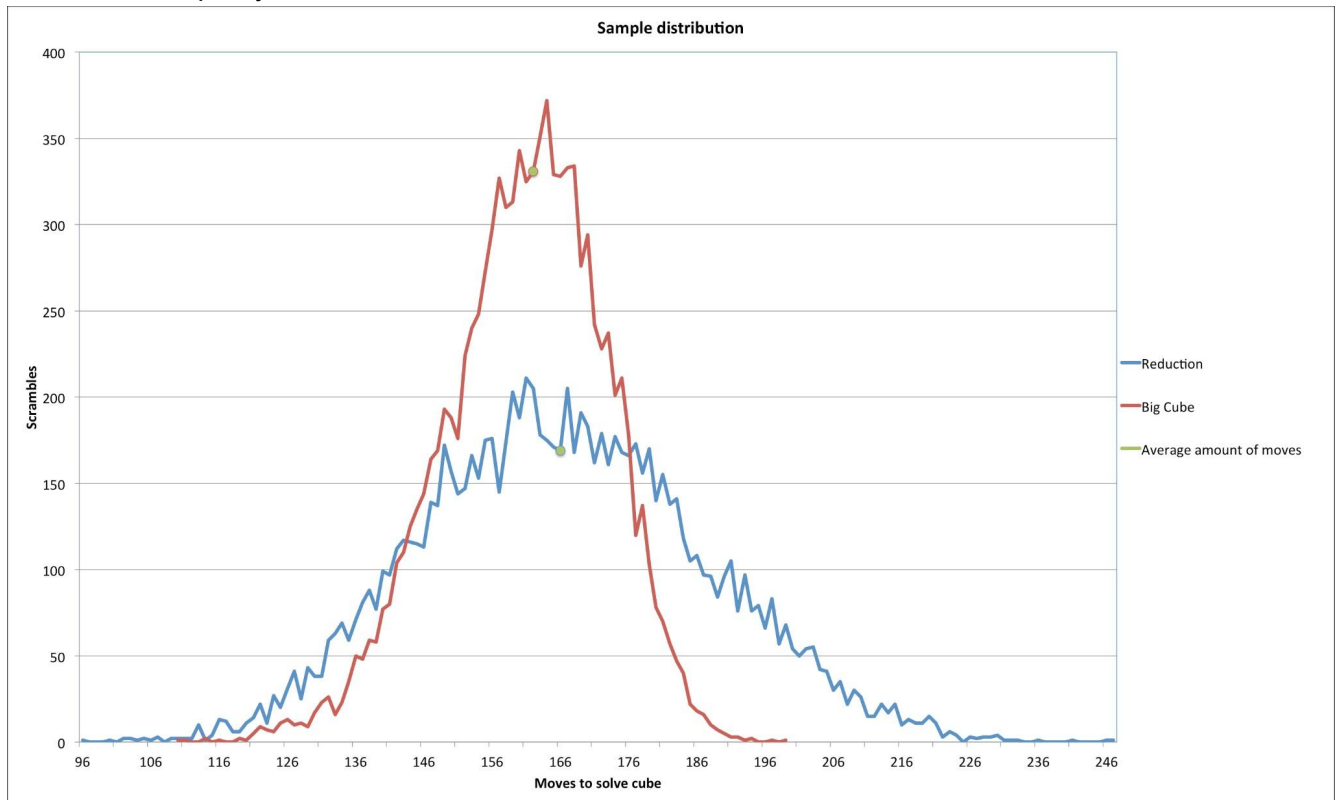
Table 4 - Reduction parity data

	<b>Total</b>	<b>Prob</b>
<b>Paritys</b>	7642	76%
<b>Single parity</b>	5026	
<b>Double parity</b>	1308	
<b>PLL</b>	5134	51%
<b>Oll</b>	2508	25%

In table 1 we can see that the Big Cube solver had an average number of moves of about 160 compared to the Reduction solver in table 3 that had an average of about 166. We measured the total number of moves as well as the number of moves in each sub-step of the solutions.

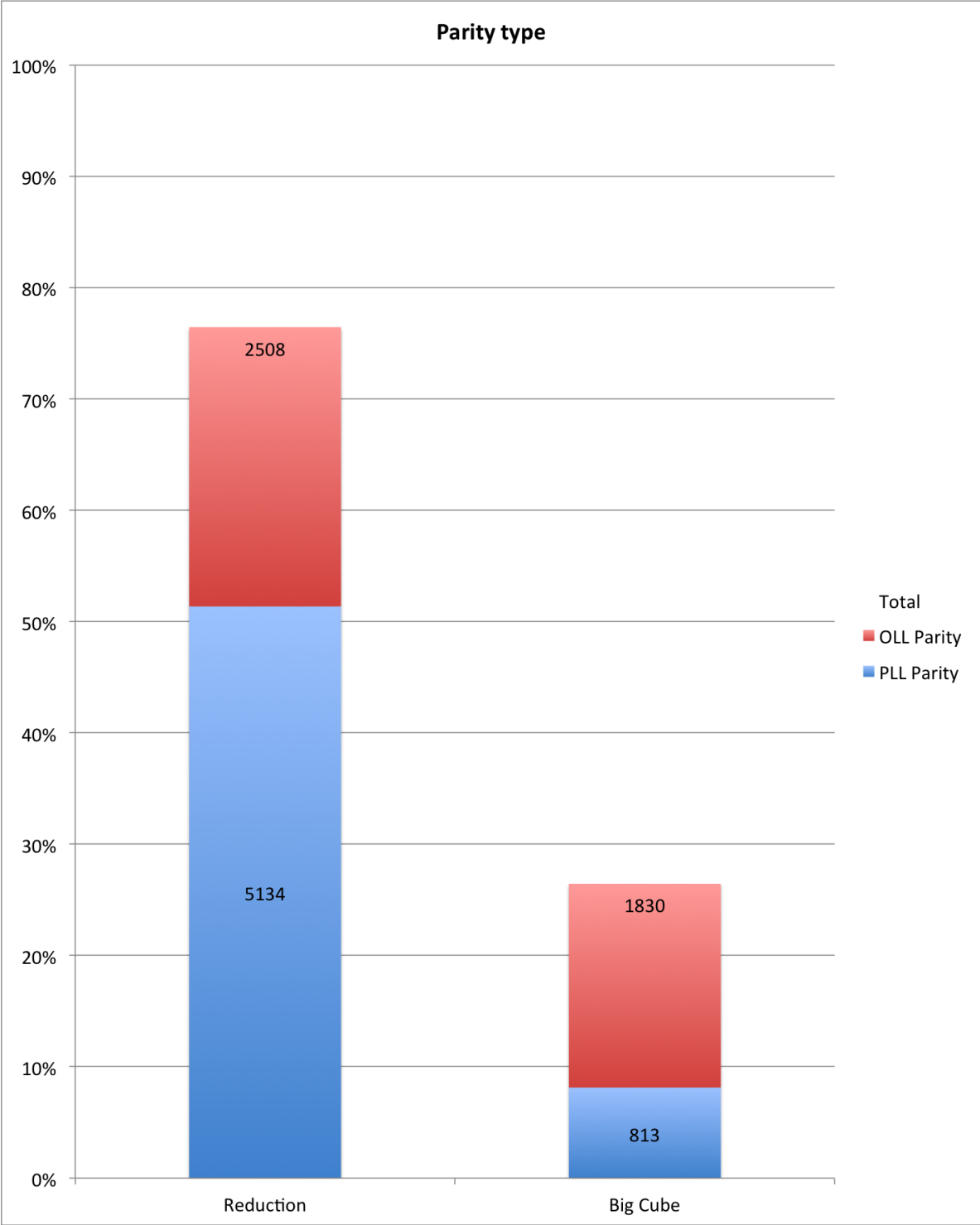
In tables 1 and 3 we can see the average number of moves in each sub-step. These sub-steps can be seen in the Solving methods section. Here we can also observe the maximum and minimum number of moves (the extreme cases).

In tables 2 and 4 we can see the number of parities each solver encountered. We can see that the Reduction solver had to solve parities 76% of the scrambles while the Big Cube solver only had to solve a parity 26% of the scrambles.



**Graph 1 - Test sample distribution (See appendix A for full size graph)**

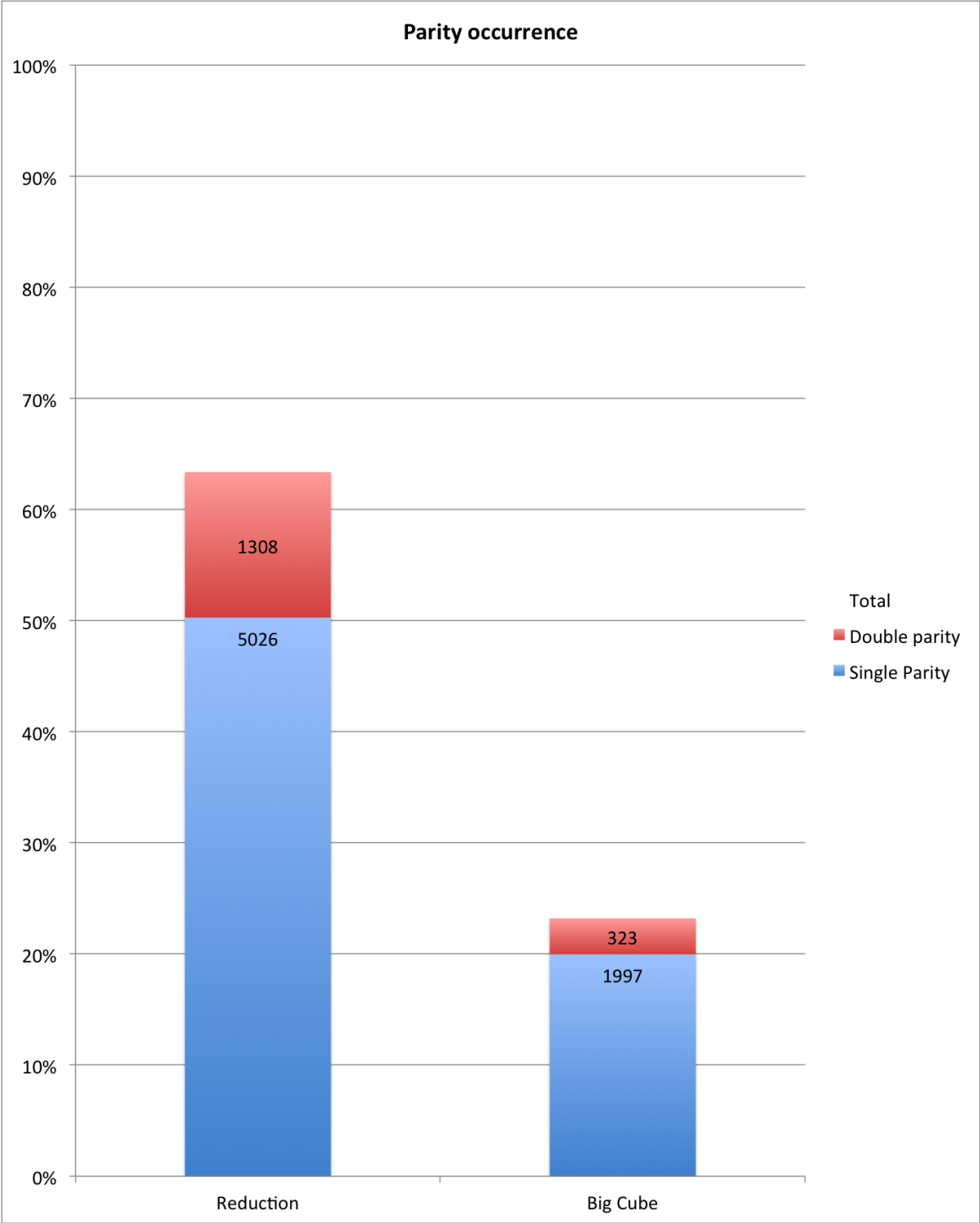
In Graph 1 we can see that the distribution curve for the Big Cube solver is much steeper indicating a more stable solving method. The Reduction solvers spread is much bigger both ways. The Reduction solver holds both the lowest and the highest amount of moves for a solve.



Graph 2 - Parity types

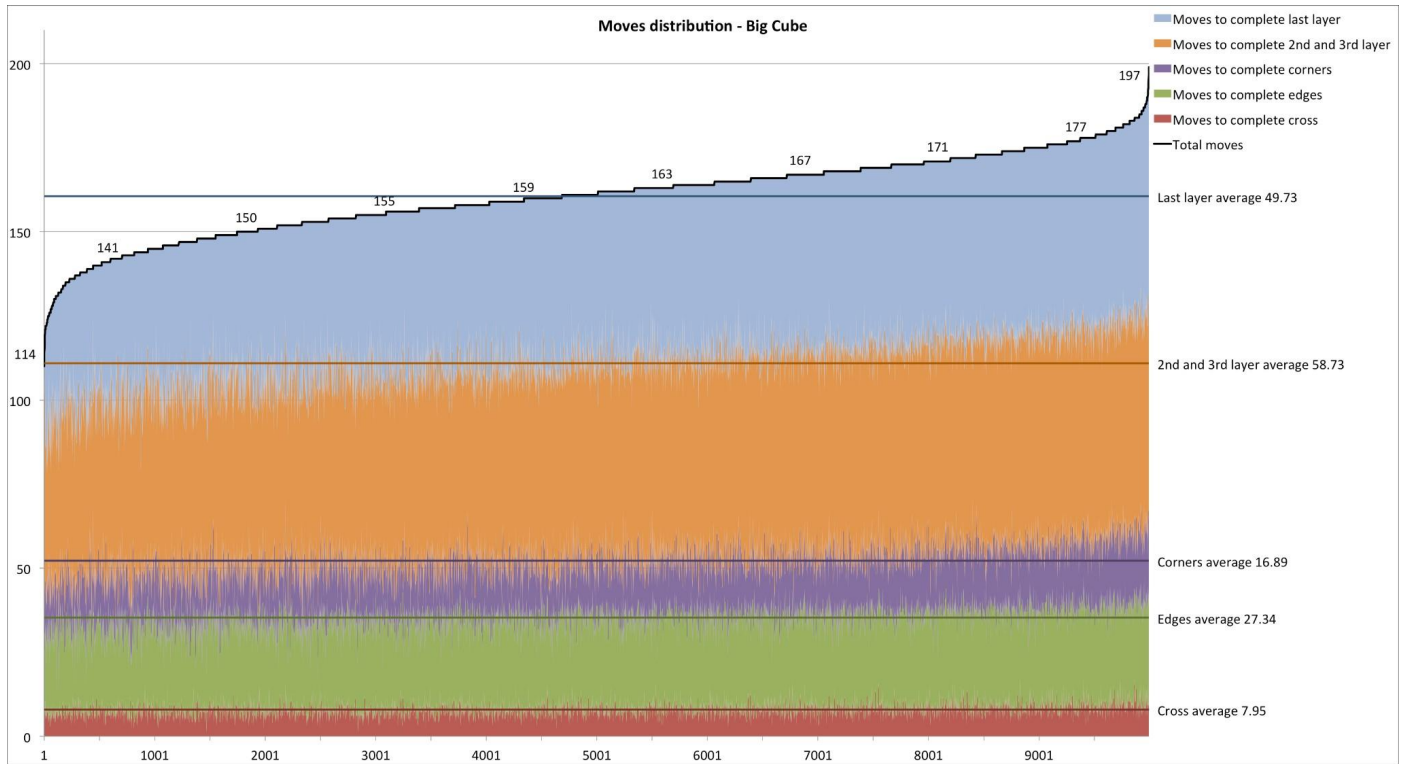
In Graph 2 we can see the how often the different parities occur. As we can see the PLL:OLL ration is almost inverted between the solvers.



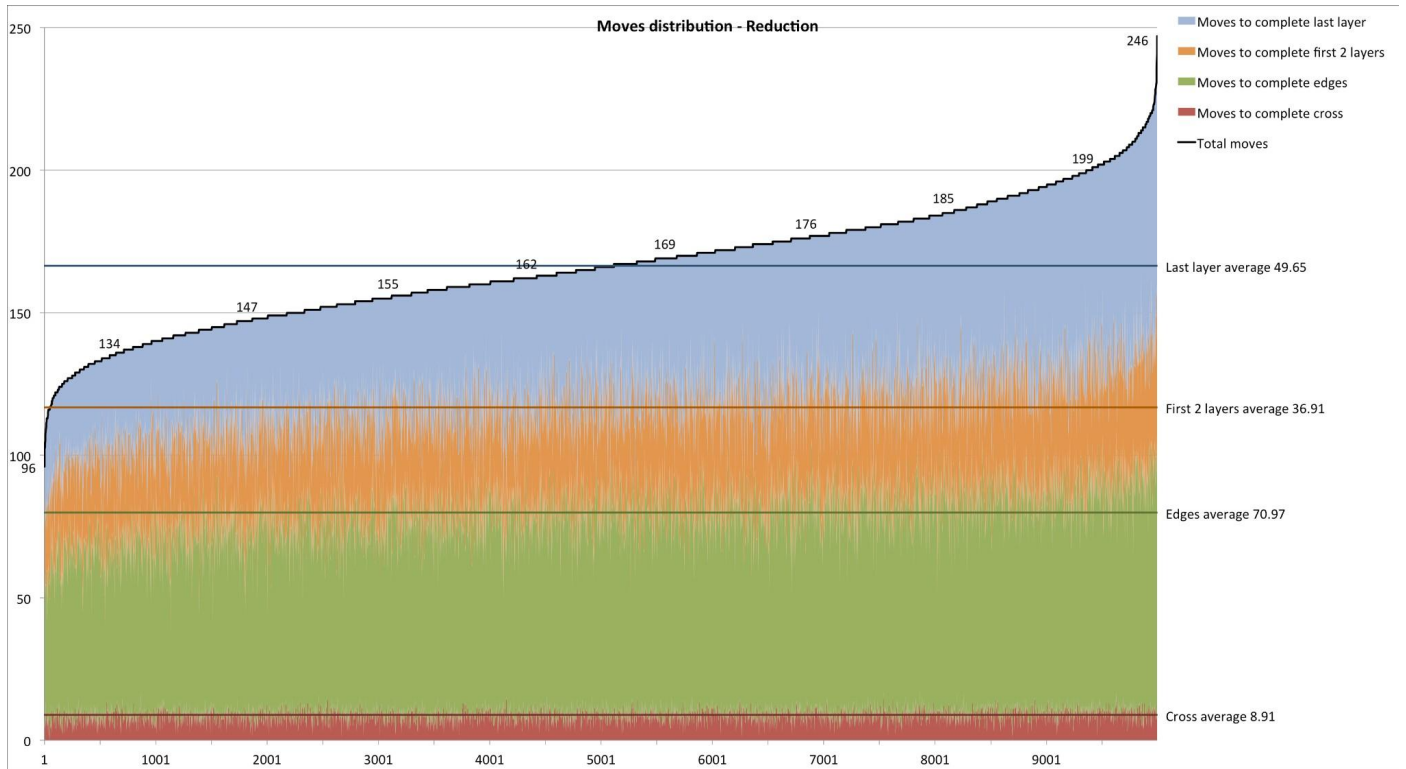


Graph 3 - Parity occurrence

In Graph 3 we can see that in most cases only one of the parities occur. *Double parity when both types occur during a solve and Single parity when only one of the parities occur.*

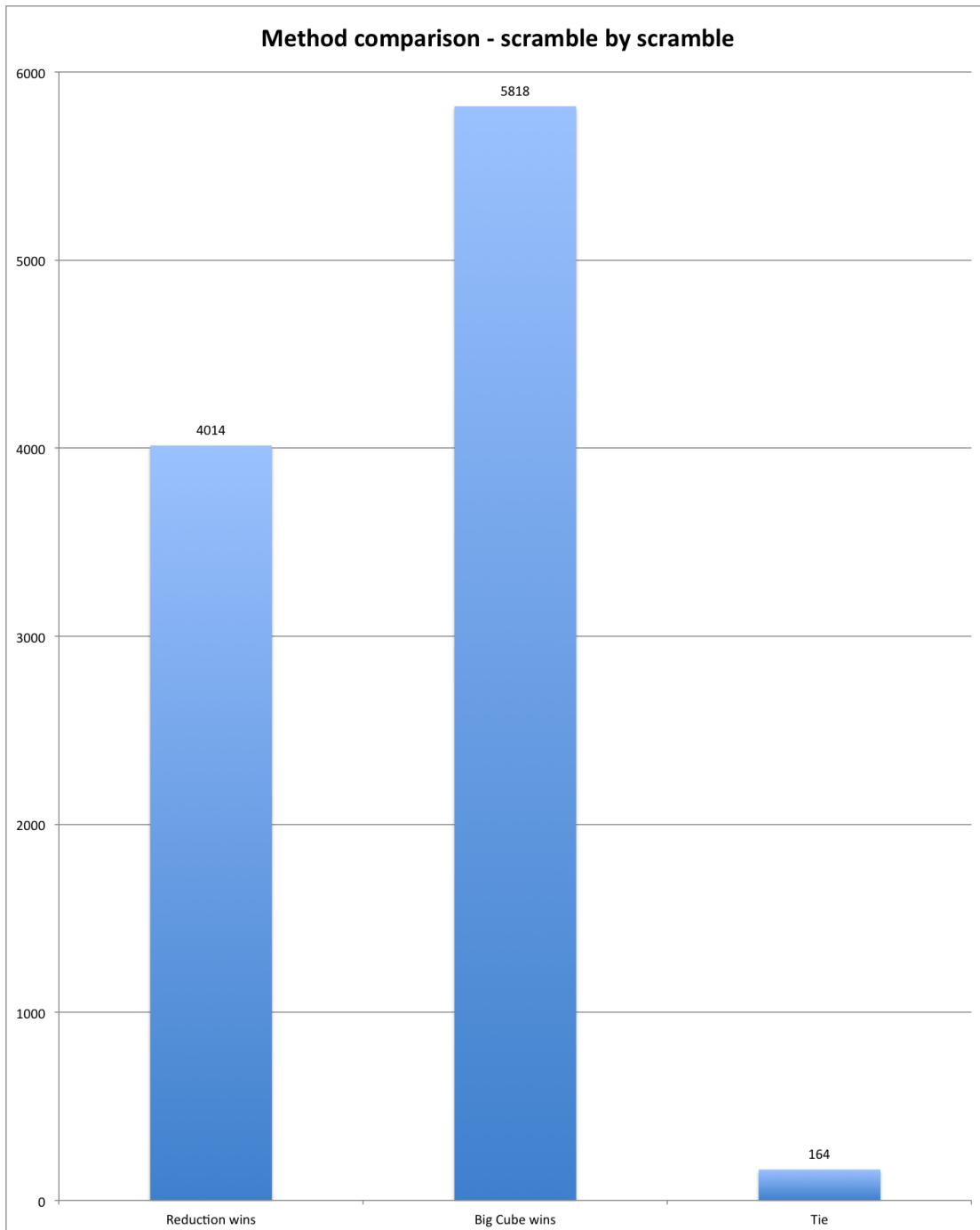


Graph 4 - Moves distribution Big Cube (see Appendix B for full size graph)



Graph 5 - Moves distribution Big Cube (see Appendix C for full size graph)

In Graph 4 & 5 we can see how the move count is distributed amongst the sub-steps within each solver. The graphs are sorted from the best solution (to the far left) to the worst solution (to the far right). We can also see the average number of moves within each sub-step.



Graph 6 - Method comparison scramble by scramble

In Graph 6 we can see how many times a solver solves the cube with less moves than the other solver. The Big Cube solver wins over the Reduction solver in almost 60% of the test cases.

# Discussion

## Move expectancy and variation

In the results we can see that the Big Cube solver has a winning percentage of about 60% over Reduction. The average of the Big Cube solver is about 160 moves versus the Reductions average of about 166 moves. Furthermore the Reductions highest solve is 247 moves compared to Big Cube's 199 maximum move count. The Reduction method beats the Big Cube method on one value, and that is the minimum move count of 96 against Big Cubes 110.

According to these results the Big Cube solver is in most cases the better method. However, the Reduction solver has a lower minimum move value and a higher maximum value. As we can see on Graph 1 and in tables 1.1 and 2.1 the Reduction solver has a higher standard deviation. This results in higher fluctuations in number of moves required to solve the cube. If these scrambles were to represent competitions the Big Cube solver would have won the most competitions but the Reduction solver would have the world record. Using a method with higher move variation is more of a gamble and could go either way. You could say that the methods serve different purposes.

When looking at the sub-step moves in graph 4 & 5 we can see what sub-steps that take up most of the move count in the solvers. The last layer's move count and the cross is almost the same for both methods. So if we discard those we see that the biggest move count within the Big Cube solver comes from solving layers 2 & 3. This step has an average of about 58 moves and a maximum of 64 moves. This means that the move count is almost always at its maximum (5011 cases actually). A very reliable method in other words. If we look at the Reduction method however the highest sub-step move count lies within the edges step. And this step varies from 29 to 99 moves. This opens up for a hybrid method.

The edge step comes right after the cross step (in the Big Cube method) and the cross step is actually the same as the beginning of the edge step in the Reduction method (see the steps in the Solving methods section). Therefore the best approach would be to see how many edgepairs that are already completed upon entering the edge step. If many are paired up, one would benefit from continuing with the Reduction method. On the other hand, if many edges are unpaired, you go with the Big Cube method. This enables a certain level of control over the fluctuations caused by the Reduction method. If one could master both methods one would get the best of both solutions, the stability of the Big Cube and the low move count deviations of the Reduction method.

## Parity

To fix parity errors requires a costly algorithm; the worst parity adds an extra 18 moves in order to solve the cube. In Graph 2 we can see that the Reduction method deals with parity errors far more often than the Big Cube method. The parity errors are easily recognized when attempting to solve the last layer. If it were possible recognize them earlier in order to prevent them from

occurring, this would make the Reduction solver more successful. However we were unable to draw any conclusions on this subject from our tests and research.

## **Complexity of the methods**

There is another element to account for since these methods are meant to be used by competitors. The number of algorithms needed in order to solve the cube using the Big Cube method (optimally) significantly exceeds the number of algorithms needed to solve the cube using Reduction. Memorizing the necessary algorithms becomes difficult. In addition most competitors already know the algorithms for the 3x3x3 cube (which is half of the steps in the Reduction method) before learning the 4x4x4 cube.

## **Sources of error**

We have only measured the number of rotations used to solve the cube; we have treated all the moves as equally costly. In reality a move that moves the bottom inner layer is probably harder to perform than a move that moves the right outer layer.

This is somewhat mitigated by the fact that many algorithms flip the cube into position so that the moves become easier to perform.

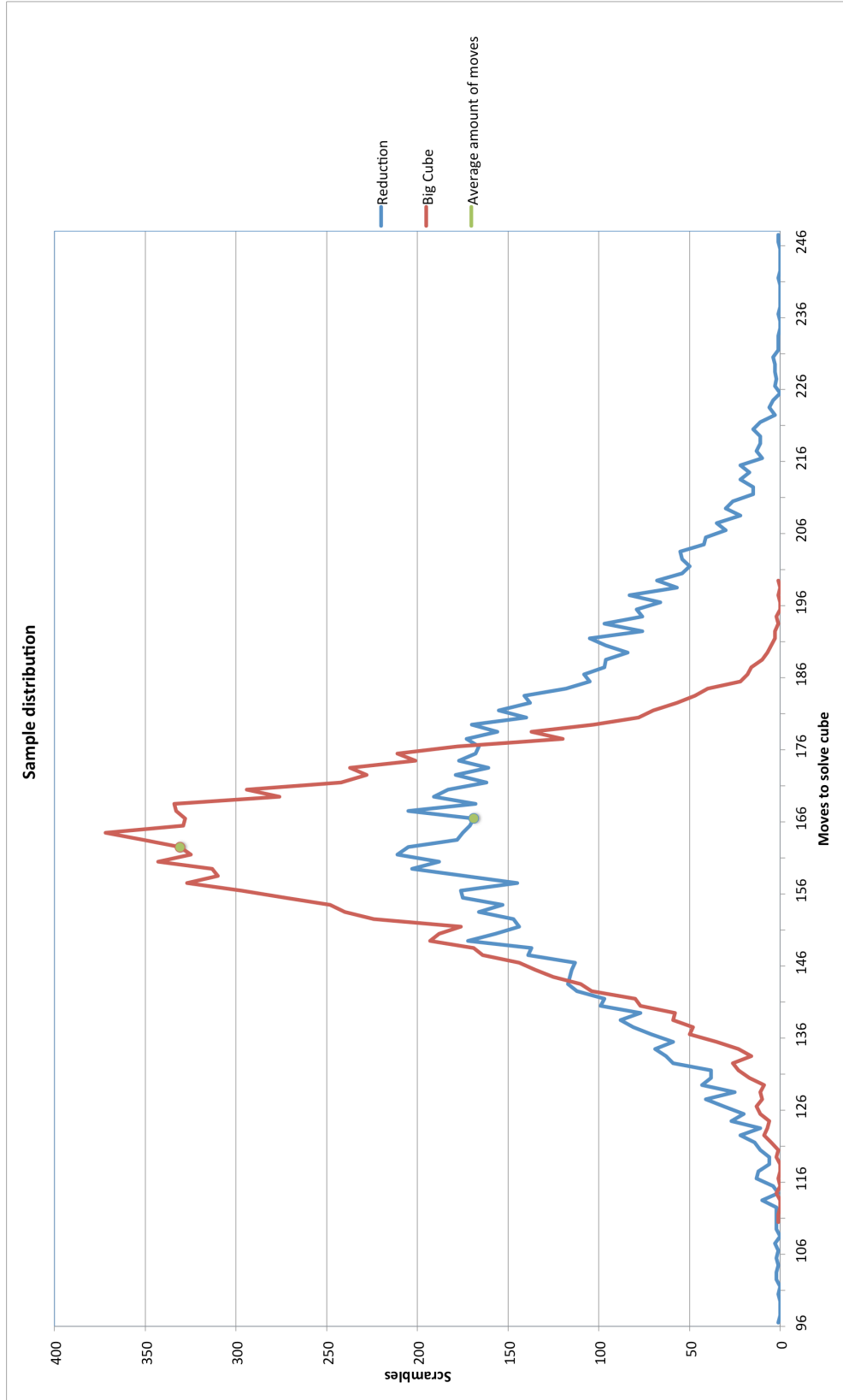
These assumptions affect the outcome of our results. How much they affect them is very hard to determine. All competitors have different preferences so taking into account how much harder one move is in regard to another would require extensive empirical studies. Our testing environment is only a portion of what goes on in reality, there are so many factors that are not feasible to implement.

The code written to implement these algorithms is massive and complex. We have written and tested around 7500 lines of code during 4 weeks so there could be errors that haven't shown during our tests. The cube implementation has been proven to some extent though (it successfully scrambles 9996 scrambles). And the solvers successfully solve said scrambles. Somewhere between the translations of human thought to computational calculations another portion of reality is most certainly lost. But again, we cannot take into account all the factors that weigh into a human solving of the cube.

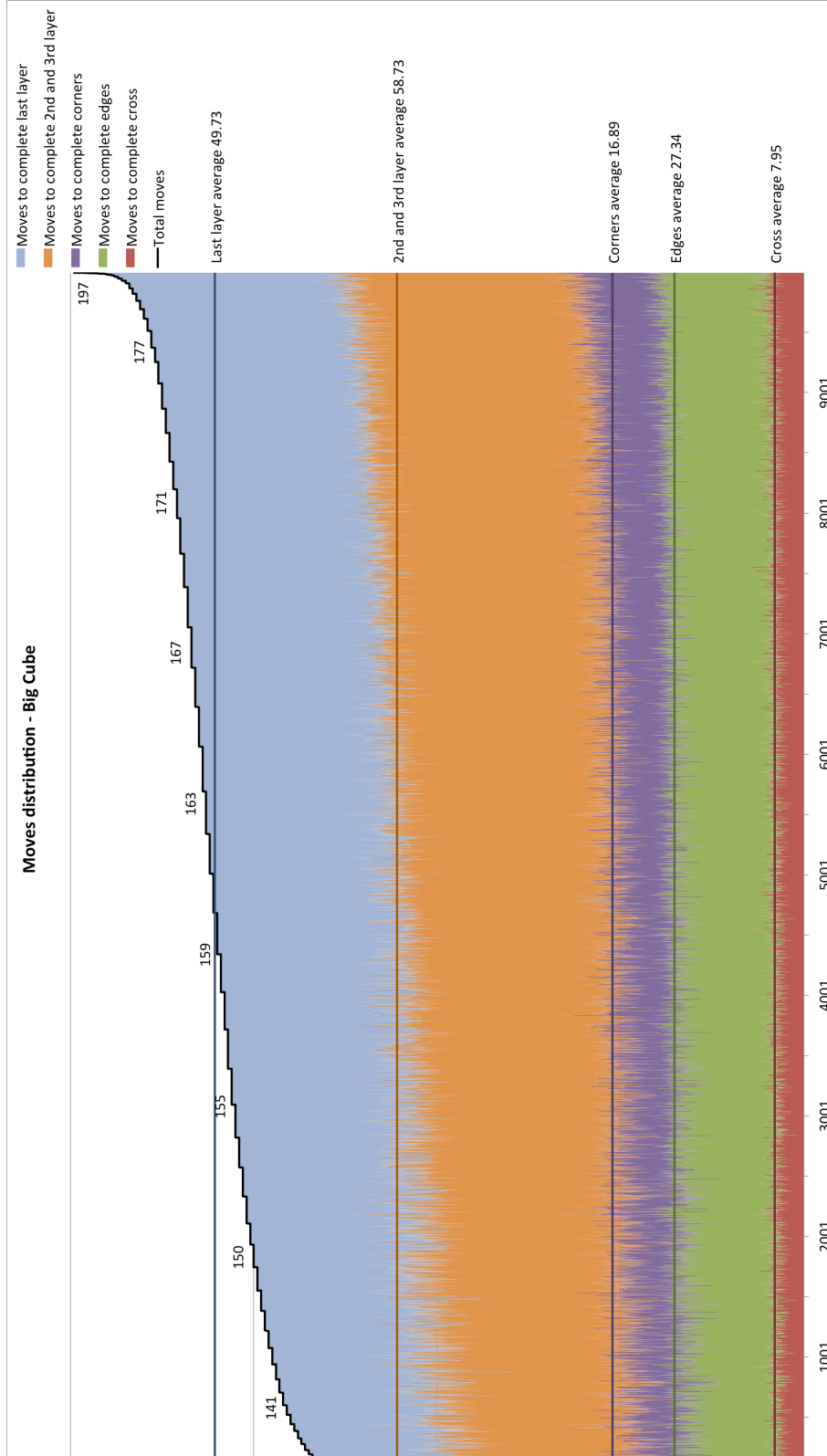
## **Conclusions**

The tests show that the Big Cube method is more successful at solving the cube in average. It is much more stable and predictive. The Reduction method has a larger span of variety in its move count. The lowest move count was 96 moves and performed by the Reduction solver. Our conclusions are that the Reduction method would result in more competition records but the Big Cube would win more competitions in total. The optimal solution would be to combine both methods as described in the discussion. That would result in a stable solution with the off chance of a low count solve.

# Appendix A



# Appendix B



# Appendix C

