BSc Thesis Applied Mathematics

Solving the 5x5x5 Rubik's cube: heuristic approach

Xander Heij

Supervisors: Jan Willem Polderman & Tom van Dijk

July, 2023

Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science

UNIVERSITY OF TWENTE.

Preface

I have been interested in Rubik's cubes for several years. And did an attempt to make a 3x3x3 Rubik's cube solving robot as my final project in secondary school. For my Bachelors Assignment, I continue studying these Rubik's cubes, but this time a more complex variant: the 5x5x5 Rubik's cube. There has not been much scientific research done about this cube, although there do exist some efficient methods to solve it, but most known information is from unpublished papers and hobby websites. I will attempt to propose my own heuristic algorithm, to solve a 5x5x5 Rubik's cube. And I will make an attempt to state something about the correctness and efficiency, in terms of move count, of the proposed heuristic algorithm.

Solving the 5x5x5 Rubik's cube: heuristic approach

Xander Heij*

July, 2023

Abstract

In this paper we present a new heuristic approach to solve the 5x5x5 Rubik's cube. First the necessary preliminary information is explained, then the permutation group of the 5x5x5 Rubik's cube is defined, and the 5x5x5 Rubik's cube is modeled mathematically. The key subsets and subgroups of the permutation group are defined, and the proposed heuristic is formulated. And an attempt is made to state something about the correctness and efficiency of the proposed heuristic algorithm. An attempt is made to compare the proposed heuristic algorithm to an efficient 5x5x5 Rubik's cube solve algorithm. By modeling the 5x5x5 Rubik's cube in python, and implementing the proposed algorithm.

Keywords: Rubik's cube, Permutation Group, Heuristic

1 Introduction & Colloquial problem statement

The Rubik's cube is a well known puzzle, and many people are able to solve it. A more complex variant of this puzzle is the 5x5x5 Rubik's cube, this puzzle is less known. And therefore also less research about it has been done. There exist several methods to solve a regular 3x3x3 Rubik's cube, some more efficient than others. And there has been determined a hard limit on the maximum required amount of moves to solve a 3x3x3 Rubik's cube. This is called God's number [9], and it has been proven that it is 20. There has not yet been determined such a number for the 5x5x5 Rubik's cube, and a lot less is known about it. There are methods to solve this cube, but a lot of improvement can be made on their efficiency. So the problem left to solve, is to come up with a method to solve the 5x5x5 Rubik's cube from an arbitrary state, to a solved state where every face exists only of pieces of the same color. And 'solving' means, finding a list of rotations, called 'moves', that would turn the cube from the arbitrary state to a 'solved' state. The challenge is to do this in as few moves as possible. This leads to the following research question:

> In what way can a 5x5x5 Rubik's cube be efficiently solved from an arbitrary state?

The needed preliminary information and notation about the 5x5x5 Rubik's cube is explained in section 2. The general notation about the parts of the cube, the notation for the moves, and the composition of moves are described. In section 3, the 5x5x5 Rubik's cube will be modeled by defining a state, the permutation group, the metric to count moves, and the size of a permutation. The problem will be stated again in section 4, but then formally, using on the the introduced notation from section 2 and 3. In the next 2 sections

*Email: a.c.heij@student.utwente.nl

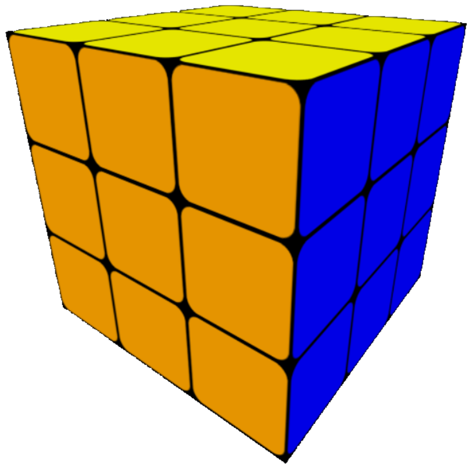

FIGURE 1: 3x3x3 Rubik's cube

is explained why the proposed algorithm must be a heuristic algorithm, and which subgroups and subsets are needed. The heuristic algorithm is introduced in section 7, and its correctness is discussed in section 8. It has been attempted to analyze whether the proposed algorithm is efficient compared to existing solutions. Which leads to a conclusion, and recommendations for future work.

2 Preliminaries

2.1 Rubik's cubes

A regular 3x3x3 Rubik's cube is a puzzle existing of 3x3x3 smaller cubes called 'cubies'. The cube has 6 sides, each side is called a 'face'. Each face can be rotated, and after a turn of 0, 90, 180 or 270 degrees, another face can be rotated independently of the previous face. [6]

Each face lies on 3x3 cubies, so for each turn 9 cubies change position. It is easy to see that however the cube would be turned, the cubie in the core, the 3 dimensional center, of the cube will always remain in the same position, therefore physical cubes do not have this core cubie, but only cubies who lie with at least 1 face on the outside of the cube. The cubies that exist on a physical cube are the pieces of that cube. There are 3 different types of pieces, center pieces, edge pieces, and corner pieces. There are 6 center pieces, these lie in the center of a face, and only have 1 visible side. There are 8 corner pieces, each corner lies on the intersection of 3 faces, therefore it has 3 visible sides. There are 12 edge pieces, one in between each pair of corner pieces. Each edge piece lies on the intersection of 2 faces, and has 2 visible sides.

Each visible side of a piece has a different color, the colors are assigned to the pieces in a way that each face of a cube in the solved state (factory state) has the same color. The colors in a solved state are White opposite of Yellow, Orange opposite of Red, Green opposite of Blue. When white is on top and orange is on the left then green is in front. The short notation of these colors is the starting letter (W, Y, O, R, G, B). The visible, colored, faces of a piece are called 'facelets'. The facelets can also be seen as the stickers of a physical Rubik's cube.

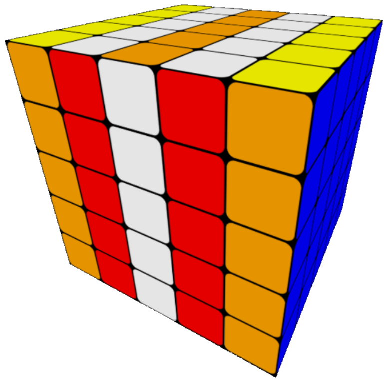

FIGURE 2: 5x5x5 Rubik's cube

When a face is rotated, the pieces of the Rubik's cube lying on that face rotate around. This means that the facelets change position, and therefore the colors of the facelets of each face can change. A cube in an arbitrary state is called 'scrambled', and the process of rotating faces of the cube arbitrarily to mess up the colors is called 'scrambling'. The challenge of a Rubik's cube is to start with a scrambled cube and to find the rotations to bring the cube back in its original, solved, state, where all facelets of each face have the same color.

A 5x5x5 Rubik's cube is a more complex variant of the normal, 3x3x3, Rubik's cube. This cube has 5 layers in each axis that can be turned independently from each other. Each face has both an inner and outer layer. For an NxNxN cube each layer is denoted as Nth outer layer, but a 5x5x5 cube only has an outer layer and a 2nd outer layer, so in this case the 2nd outer layer is called 'inner layer'. See for example the inner layers of the left and right face in Figure 2 with red in front and white on top. All NxNxN cubes larger than a 3x3x3 are called 'big-cubes'.

A 5x5x5 Rubik's cube has a 3x3 center on each of the 6 faces, each 3x3 center consists of a center, 4 corners and 4 edges. Therefore the following notation will be used:

- Center-center: The middle of the 3x3 center.
- Corner-center: A corner center piece.
- Edge-center: A edge center piece, in between two corner-centers.

An example can be seen in Figure 3. The green facelets on the left side are corner-centers. The orange facelets on the front side are edge-centers. And the yellow center facelet on the up side is the center-center.

2.2 Move notation

There are multiple ways to denote the turns you can make on a physical Rubik's cube, but the most commonly used notation is the Singmaster notation [10]. Each face of the cube is named, and a move on that face is called accordingly. The names of the faces are: Left, Right, Front, Back, Up, Down. These names are chosen in a way that each starts with a different letter, these letters are used to describe the moves that turn the faces. For example, a turn of the left side of the cube is called 'L', a turn of the top side is called 'U'. Each side of a physical cube can be turned either 0, 90, 180, or 270 degrees without

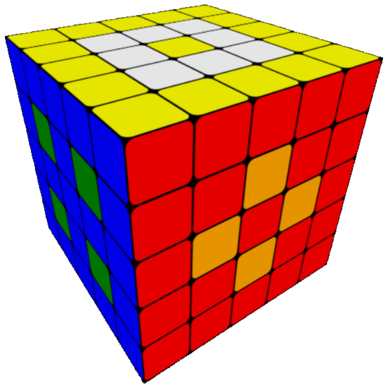

FIGURE 3: Centers of the 5x5x5 Rubik's cube

blocking another move on another axis.

A single move describes a clockwise turn of 90 degrees of a certain face of the cube. A short form for a turn on a face of 180 degrees is the letter of that face followed by the number '2'. For example a 180 degree turn of the front side of the cube can be written as $F2$. A clockwise turn of 270 degrees is equivalent to a counterclockwise turn of 90 degrees, but the latter is more time efficient on a physical cube, and therefore this is always in literature. The notation of a counterclockwise turn of any face is denoted as the letter of that face followed by a prime, or the letter is raised to the power of minus one. Examples of a counterclockwise turn of the right face are: R' and R^{-1}. In this paper the prime notation is used.

2.2.1 Slice moves

There are also names for the middle layers of the cube, even though these moves are not strictly necessary. Since a middle layer turn is equivalent to turning both outer layers on the same axis. But these moves will be used later in this paper, so they need to be properly defined. A turn of a middle layer is called a slice turn. A 3x3x3 Rubik's cube has 3 slices called: Middle, Equator, and Standing. These are the slices that exists on every odd NxNxN Rubik's. The notation for the other slices of big-cubes will be explained in the next subsection.

An 'M' move turns the slice between the L and R faces, seen from the L face. This means that M is equal to the two moves L and R' combined. The 'E' move turns the slice between the U and D faces, seen from the U face. So E is equal to U and D' combined. The move S turns the slice between the F and B face, seen from the F face. So S is equal to F and B' combined. An example of slice moves can be seen in Figure 4.

2.2.2 Big-cube moves

The combination of slice moves and outer moves leads to the following set of single moves that can be performed on a 3x3x3 Rubik's cube:

$$\{\mathrm{U}, \mathrm{U}', \mathrm{U2}, \mathrm{D}, \mathrm{D}', \mathrm{D2}, \mathrm{L}, \mathrm{L}', \mathrm{L2}, \mathrm{R}, \mathrm{R}', \mathrm{R2}, \mathrm{F}, \mathrm{F}', \mathrm{F2}, \mathrm{B}, \mathrm{B}', \mathrm{B2}, \mathrm{M}, \mathrm{M}', \mathrm{M2}, \mathrm{E}, \mathrm{E}', \mathrm{E2}, \mathrm{S}, \mathrm{S}', \mathrm{S2}\}$$

But in this paper we are interested in the 5x5x5 cube, this puzzle has more layers that can be turned, the moves that define those turns needs to be defined as well.

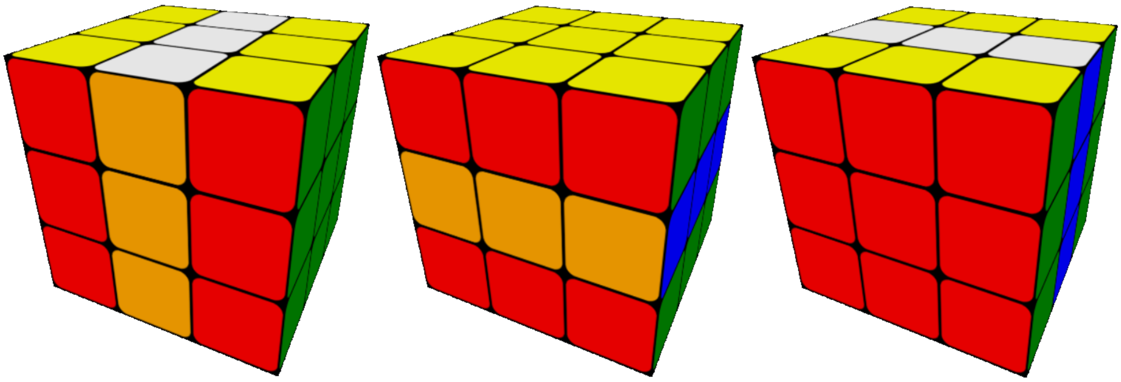

FIGURE 4: Moves M2, E2 and S2 performed on 3x3x3 Rubik's cube

When the number N is put in front of a move which turns a certain face, this face will be turned in the Nth outer layer of the cube. For a move of the outer layer, it is not needed to add the number '1' before a move. So for example the move 2L will turn the inner layer of a 5x5x5 Rubik's cube, while the move L will turn the outer layer. When 'w', which stands for 'wide', is added after a move, then the first until Nth outer layer will be turned simultaneously. So for example the move 2Fw turns both the inner and outer front layer of a 5x5x5 Rubik's cube. The 'w' is added before a '2' or a prime, so the move that describes the turn of both the inner and outer front layer over 180 degrees is denoted as 2Fw2.

2.3 Move composition

A sequence of moves is called an 'algorithm'. The moves in this sequence should be performed from left to right. This is counterintuitive from an algebraic point of view, since moves are elements in a group, and composition is the group operation. Which will be explained in more detail in a later section. Normally composition works from right to left. In for example function composition, $g \circ f = g(f(x))$ which means that f needs to be applied first, and the result needs to be applied on g. For moves of a Rubik's cube it is the opposite. For example $F \circ R$ describes a front move followed by a right move.

The short notation of $p \circ q$ is pq where both p and q are moves. Many moves defined above are equal to a composition of other moves. A double turn is equal to the composition of 2 single turns. And a counterclockwise turn is equal to 3 clockwise turns. $pp = p^2 = p2$ and $ppp = p^3 = p'$ for all $p \in \{\mathrm{U}, \mathrm{D}, \mathrm{L}, \mathrm{R}, \mathrm{F}, \mathrm{B}\}$.

3 Modeling the 5x5x5 Rubik's Cube

3.1 State and individual facelets of a 5x5x5 Rubik's cube

The state of a 5x5x5 Rubik's cube can be denoted by the permutation of its facelets. A 5x5x5 Rubik's cube has 6 faces with 5x5 facelets, every state of the cube corresponds to a permutation of these facelets. Therefore the state of a 5x5x5 Rubik's cube is an element of S_{150}, the permutation group on 150 symbols. The 150 facelets can be denoted as a two dimensional coordinate (c, i), where c denotes the color of the facelet, and i denotes the index of the facelet ranging from 0 to 24. So $(Y, 12)$ is the center-center of the yellow face for example. The set of all facelets is be denoted by S.

3.2 The permutation group of the 5x5x5 Rubik's cube

Not all permutations of the facelets in S_{150} are reachable states of the 5x5x5 Rubik's cube, due to the mechanical limitations. The set of all possible states is a subset of the state set S_{150}. A permutation of the facelets is a possible state if there exists a sequence of moves which turns the solved cube in this state. The solved element, where all 25 facelets of each face are the same color, is denoted as I. The permutation group of all reachable states can be defined in the following way:

Definition 3.1 $G(5) = \{\phi \mid \phi : S \to S,\ \phi = \phi_0 \circ \phi_1 \circ \cdots \circ \phi_n$ *for which* $\psi_i \in \{\mathrm{U}, 2\mathrm{U}, \mathrm{D}, 2\mathrm{D}, \mathrm{L}, 2\mathrm{L}, \mathrm{R}, 2\mathrm{R}, \mathrm{F}, 2\mathrm{F}, \mathrm{B}, 2\mathrm{B}\}\}$

For future reference, the set $\{\mathrm{U}, 2\mathrm{U}, \mathrm{D}, 2\mathrm{D}, \mathrm{L}, 2\mathrm{L}, \mathrm{R}, 2\mathrm{R}, \mathrm{F}, 2\mathrm{F}, \mathrm{B}, 2\mathrm{B}\}$ is denoted as $\mathrm{Gen}(G(5))$, because this set forms the generator of $G(5)$.

Notice that each element of the group can be viewed as both a permutation of the facelets of the cube, and a state of the cube itself. Since each permutation performed on the solved state results in a state corresponding to that permutation.

Since all states can be reached by the composition of single turns, $G(5)$ is generated by $\mathrm{Gen}(G(5))$. This is not the smallest generating set of $G(5)$, but it is the most intuitive. It is not needed to add the middle slice in the generator since turning the middle slice is equivalent to turning all outer and inner slices of the faces on the same axis. But it is actually also not needed to add moves for turning 1 of the 6 faces in the generator. This has been proven by Roger Penrose, and is written down in the notes about the 3x3x3 Rubik's cube by David Singmaster [10].

3.3 Equivalent states

The defined group does not completely model a physical cube, since the equivalent pieces have not been taken into account. Unlike the 3x3x3 Rubik's cube for which all pieces are unique, does the 5x5x5 Rubik's cube have similar pieces. The facelets of these pieces have the same color and therefore the pieces can't be differentiated from each other on a physical cube. The existence of these equivalent pieces lead to equivalent states. 2 states are considered to be equivalent if they look exactly the same on a physical Rubik's cube, which means that the colors of the facelets on each face are the same for both states.

The following equivalence relation can be defined:

Definition 3.2 *let* $\phi, \psi \in G(5)$ *be 2 states of the 5x5x5 Rubik's cube.*
$\phi \sim \psi$ *if* $c(\phi(s)) = c(\psi(s))\ \forall s \in S$

It can be easily verified that this definition follows all the properties of an equivalence relation.

- $\phi \sim \phi\ \forall\ \phi \in G(5)$. Since $c(\phi(s)) = c(\phi(s))\ \forall\ s \in S$.
- $\phi \sim \psi \implies \psi \sim \phi\ \forall\ \phi, \psi \in G(5)$. If $c(\phi(s)) = c(\psi(s))$ then $c(\psi(s)) = c(\phi(s))$ for all $s \in S$
- $\phi \sim \psi$ and $\psi \sim \chi \implies \phi \sim \chi\ \forall\ \phi, \psi, \chi \in G(5)$. Since $c(\phi(s)) = c(\psi(s))$ and $c(\psi(s)) = c(\chi(s))$ implies that $c(\phi(s)) = c(\chi(s))$ for all $s \in S$.

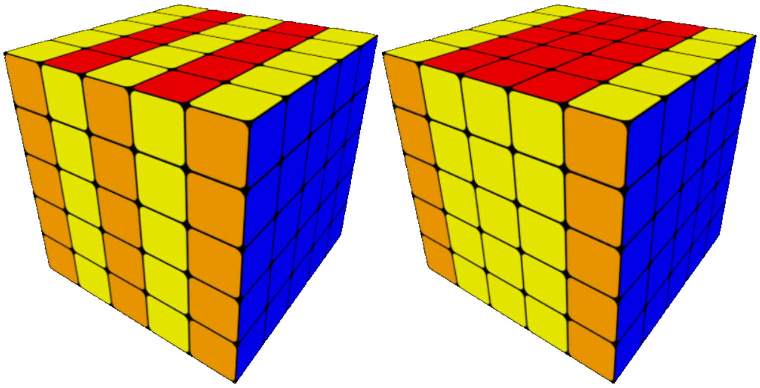

FIGURE 5: Moves 2L 2R' and 2L M 2R' performed on 5x5x5 Rubik's cube

3.4 Metrics

When looking at move-efficiency of methods to solve Rubik's cubes, first we need to be determine what counts as a single move. There are multiple metrics for this. The most common metric is HTM, which stands for Half Turn Metric. A single move in HTM is defined as a turn of a single face. This means that both a double move, and a single move of a face count as one move. So for example both U2 and U count as a single move, but LR' counts as 2 moves, since the L and the R face are 2 different faces.

The metric that will be used in this paper is STM which stands for Slice Turn Metric. A move is defined as single synchronous turn of adjacent slices in the same direction on the same axis. In STM not only U2, U but also 2Lw 2Rw' counts as 1 single move. Since 2Lw 2Rw' is equivalent to the move M, which is a move of a single slice. 2L 2R' is not a single move, while 2L M 2R' does count as a single move. This example can be seen in Figure 5. Slice turn metric is much more intuitive for cubes larger than a 3x3x3. Because it is common to turn multiple inner slices together, which would otherwise count as multiple moves. Therefore STM be the metric used in this paper.

3.5 Size of a permutation

The set of all single valid moves according to STM is a superset of Gen($G(5)$), this set is denoted as $\text{Gen}_{\text{STM}}(G(5))$.

By definition, every element in $G(5)$ can be factorised into elements of Gen($G(5)$). The fact that $\text{Gen}_{\text{STM}}(G(5))$ is a superset of Gen($G(5)$) implies that every element in $G(5)$ can also be factorised into elements of Gen($G(5)$).

Theorem 3.1 (Factorisation of elements in G(5)) *For all $\phi \in G(5)$ there exists $\psi = \psi_0 \circ \psi_1 \circ \cdots \circ \psi_n$ for which $\psi_i \in \text{Gen}_{\text{STM}}(G(5))$ such that $\phi = \psi$*

The factorisation of a permutation corresponds to the amount of moves needed to solve the state corresponding to that permutation. This is how the size of a permutation is defined:

Definition 3.3 $|\phi|$ = *minimal n such that $\phi = \psi_0 \circ \psi_1 \circ \cdots \circ \psi_n$, for which $\psi_i \in$* $\text{Gen}_{\text{STM}}(G(5))$

4 Formal problem statement

The problem is to find a method that solves the cube from an arbitrary state. This means finding a sequence of moves such that the composition of the starting state and the sequence of moves is the solved state. This can be written as: Find $\psi : G(5) \to G(5)$ such that $(\psi(g))(g) = I$ for all $g \in G(5)$. ψ maps all possible states of the cube to an permutation that solves that state. Since $(g \circ g^{-1}) = I$ for any $g \in G(5)$, the problem could be seen as finding the inverse of $g \in G(5)$. But we are not only interested in g^{-1}, which is the permutation to solve state g, the main problem is to find which moves to perform on the state g to solve it. This means finding the factorization in moves in $\text{Gen}_{\text{STM}}(G(5))$ of g^{-1}. It has been shown that such factorization must exists. It is also important that the factorization should be a rather short sequence of moves. This means that $|g^{-1}|$ should be small.

5 Heuristic

Finding an factorization of g^{-1} for a permutation $g \in G(5)$ is equivalent to solving a 5x5x5 rubiks cube starting in state g. It has been proven that finding a solution in the least amount of moves in STM to solve an NxNxN Rubik's from an arbitrary state is NP-complete [1]. So when a solution is given, it can be verified quickly, but there is no known method to come up with an optimal solution in a short amount of time.

Brute force is needed to find the optimal solution for an NxNxN Rubik's cube, which doesn't scale well as N increases. Therefore it is not feasible to find the optimal solution of a 5x5x5 Rubik's cube. There isn't even a known limit yet of the maximum amount of moves needed to solve a 5x5x5 Rubik's cube optimally. While this limit is known for a 3x3x3 Rubik's cube [9]. Therefore the proposed solution will be a heuristic algorithm, which will not necessarily find the optimal solution.

The term heuristic algorithm is not mathematically defined. But it means generally, an algorithm for a decision problem that provides quality solutions within a reasonable amount of time for which it is not guaranteed that the given solution is optimal [7]. In the context of the 5x5x5 Rubik's cube, this means that the proposed algorithm will always provide a correct solution, but not the solution which requires the least amount of moves. But the aim is still to provide a solution with as few moves as possible. Because it can be proven that there exists a systematic method to swap certain pieces which can always solve a cube from any state, but this is an inefficient method that results in too many moves [5].

The approach for the heuristic algorithm will be to reduce the set of all possible states in each step. A sequence of moves is found for any arbitrary permutation of $G(5)$ such that the corresponding state will be an element of a strictly smaller set after the sequence of moves is performed. This approach can be compared to how you can plan a route from city A to city B. You may not know the fastest possible route, but you may know certain cities that lie in between city A and city B. If for example city C, D and E lie in between city A and B, and you know the shortest route from city A to C, C to D, D to E and E to B, then you know a route from city A to B. This route is not optimal, but if points C, D and E are chosen in a good way, the route can be expected to be close enough to optimal.

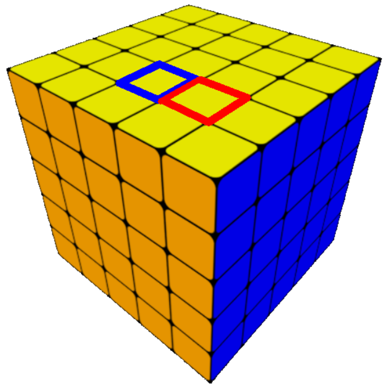

Figure 6: Up-Right-Front corner-center and Up-Front edge-center of the 5x5x5 Rubik's cube highlighted

6 Towards a heuristic

6.1 Introduction

As described in the previous section, the heuristic reduces the size of possible states in each step. For each step i, for any arbitrary state a in the set of possible states at the start of that step A_i, a sequence of moves g_i is found such that $a \circ g_i \in A_{i+1} \subset A_i$. The sets A_i need to be chosen in a clever way such that finding the sequence of moves g_i is relatively simple, and the sizes of g_i should be small. Such that the total solution, $\sum_{i=0}^{n} |g_i|$ is close to optimal. In this section, each chosen subset, or subgroup will be defined, and the size will be calculated. The heuristic consists of 4 steps.

6.2 Description of key subsets of $G(5)$

The first step will reduce an arbitrary state to an element of $K \subset G(5)$. To define subset K, first some other terms for the center pieces of the 5x5x5 Rubik's cube need to be defined.

Each corner-center piece can be referenced by 3 faces. Each piece lies on 1 of the 6 faces, each face has 4 adjacent faces, each corner center lies in the corner of 2 of these adjacent faces. The piece can be referred to by the face it lies on, and the 2 adjacent faces from which it lies in the corner. The order of the last 2 faces will be clockwise. For example, the corner-center piece on the up face, in the corner of the adjacent front, and right face, in Figure 6, can be referred as Up-Right-Front.

The same principle applies for edge-centers, they lie on 1 face, and are on the side of 1 of the 4 adjacent faces. So these pieces can be referred to by the face it lies on followed by the adjacent face. For example the edge-center on the up face on the side of the front face, in Figure 6, can be referred as Up-Front.

The subset K exists of all states for which the 3 corner-center facelets closest to each corner of the cube represent a corner piece together, and the 2 edge-center pieces closest to each edge represent an edge piece together. The 3 colors of the facelets of the corner-centers match the colors of a corner piece, and the 2 colors of the facelets of the edge-centers match the colors of an edge piece. In a way that the represented pieces together form a valid state of a 3x3x3 Rubik's cube.

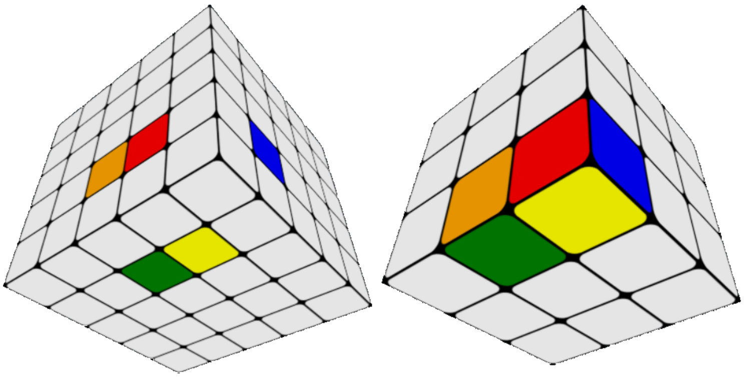

Figure 7: Example of center facelets representing pieces of the 3x3x3 Rubik's cube

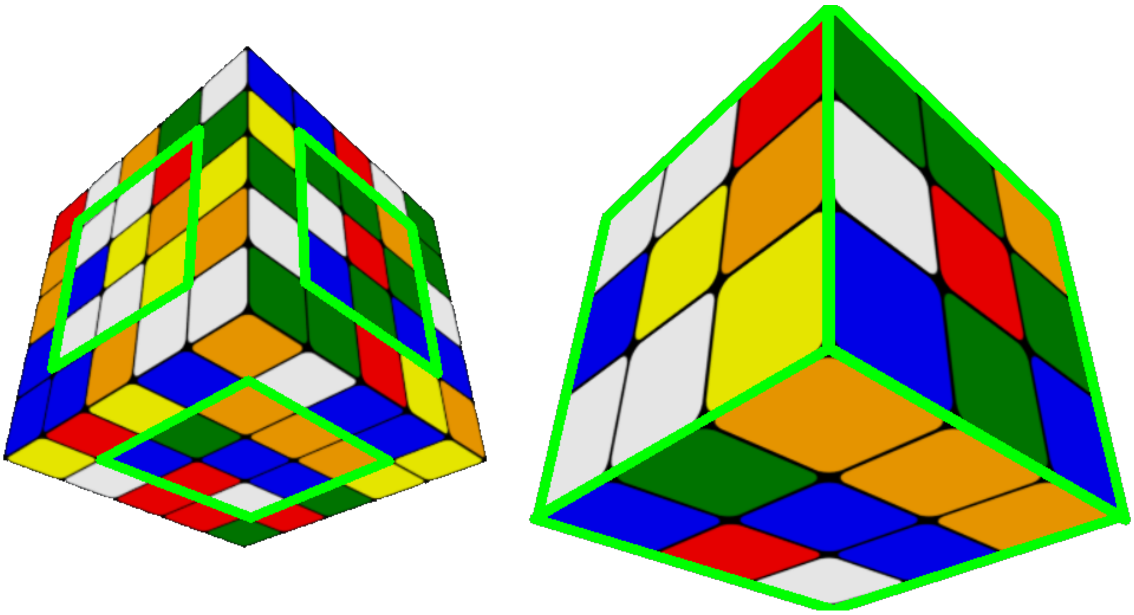

Figure 8: Example of center facelets representing a 3x3x3 Rubik's cube

So for example, if you consider the corner of the up, front and right face. The Front-Up-Right, Up-Right-Front, and Right-Front-Up center piece form a corner together. If the colors of the facelets of those pieces are respectively, yellow, red, and blue, they form the yellow-red-blue corner piece oriented with yellow to the front, red to the top, and blue to the right. If you consider the edge of the front and up face, the Up-Front, and Front-Up center piece form an edge together. If the colors of the facelets of those pieces are respectively orange and green, they form the white-green edge piece oriented with orange on top, and green in front. This example can be seen in figure 7.

Any state of the 5x5x5 Rubik's cube is an element of K when the centers of the form all the 8 corner pieces from a 3x3x3 Rubik's cube, and all the 12 edge pieces of a 3x3x3 Rubik's cube, and when they are configured in a valid 3x3x3 Rubik's cube configuration. This means that the centers of the cube are isomorphic to a state of the 3x3x3 Rubik's cube, and the outer layers can be scrambled arbitrarily.

To calculate the size of this subset, first the size of $G(5)$ needs to be calculated. There exists a general formula for the size of the group of all odd NxNxN Rubik's cubes [2], this can be used to calculate the size of $G(5)$. It is important to note that this paper is not published and not peer reviewed. I could not find a published paper proofing the correctness of this formula.

Theorem 6.1 $|G(2N+1)| = \frac{1}{2} \cdot (8! \cdot 3^7) \cdot (12! \cdot 2^{11}) \cdot (24!)^{N-1} \cdot (\frac{24!}{4!^6})^{N(N-1)}$

$|G(5)| = 282,870,942,277,741,856,536,180,333,107,150,328,293,127,731,985,672,$
$134,721,536,000,000,000,000,000.$
This is roughly $2.83 \cdot 10^{74}$.

The size of K can be combinatorially calculated in similar manner. Making use of the the fact that the configuration of the center pieces is independent from the configuration of the edge and corner pieces. This implies that the size of K is equal to the amount of different configurations of the centers and the outer layer multiplied. The configurations of the centers is equal to $|G(3)|$ since the centers of each valid state in $|K|$ is isomorphic to a state of the 3x3x3 Rubik's cube. The calculation for $|G(5)|$ uses this same fact of independence. Therefore all possible configurations of the centers can be factored out, and the total can be multiplied by $|G(3)|$ to obtain K. All possible configurations of the centers in $G(5)$ is $(\frac{24!}{4!^6})^2$. $|G(3)| = \frac{1}{2} \cdot (8! \cdot 3^7) \cdot (12! \cdot 2^{11})$. Hence the simplified calculation of the size of K is:

$|K| = (\frac{1}{2} \cdot (8! \cdot 3^7) \cdot (12! \cdot 2^{11}))^2 \cdot 24! = 1,160,695,029,268,457,213,215,565,706,414,469,$
$047,562,082,490,211,368,960,000,000,000$
This is roughly $1.16 \cdot 10^{63}$, which is $2.44 \cdot 10^{11}$ times smaller than $|G(5)|$

The second step of the heuristic will reduce an arbitrary state from subset K to an element of $L \subset K$. The subset L is a subset of K with extra restrictions for the edge pieces.

Any state in K is an element of L when all rows of 3 adjacent edges are pieces with the same color and orientation, when the center 3x3x3 representation would have been solved.

For example, the edge in between the white and blue corner-center representing the white-blue-red corner is white-red, with the white facelet adjacent to the white facelet of the corner-center, and the red facelet adjacent to the blue facelet of the corner-center. Then the edge in between the white and blue edge-center representing the white-blue edge is also white-red, with white adjacent to white, and red to blue. And the edge in between the corner-centers representing the white-blue-orange corner is also white-red, with white adjacent to white, and red to blue. When the 3x3x3 representation will be solved. the 3 adjacent edges in between the white and blue of the white-blue-red corner, white-blue edge and white-blue-orange corner, are all white-red with the same orientation.

For each edge, the position and orientation, of the two other edges with the same two colors are completely fixed by this edge. When calculating the the size of L, without loss of generality, can be assumed that the position and orientation of the two outer edges will be determined by the position and orientation of the middle edge. There are 24! possible permutations of the outer edges in the set K. Therefore the size of L is the size of K divided by 24!.

$|L| = (\frac{1}{2} \cdot (8! \cdot 3^7) \cdot (12! \cdot 2^{11}))^2 = 1,870,735,787,256,481,225,707,817,046,900,736,000,000$
This is roughly $1.87 \cdot 10^{39}$, which is $6.20 \cdot 10^{23}$ times smaller than $|K|$

6.3 Description of key subgroups of $G(5)$

The third step will reduce an arbitrary state in L to an element of the state space of $G(5)_{\text{out}} < G(5)$. The subgroup $G(5)_{\text{out}}$ is generated by $< \text{U}, \text{D}, \text{L}, \text{R}, \text{F}, \text{B} >$. Each permutation p in this group, can be solved by only moves from the set $\{\text{U}, \text{D}, \text{L}, \text{R}, \text{F}, \text{B}\}$, which implies that all the 3x3 centers of each face, and the 3x1 edges of each face have the same color(s). Since they can never be scrambled in a way that the colors differ using only moves from the generator, since only the outer layers can be turned. This means that this subgroup is isomorphic to $G(3)$ the group of all permutations of a 3x3x3 Rubik's cube.

$G(5)_{\text{in}}$ is generated by $< \text{2Uw}, \text{2Dw}, \text{2Lw}, \text{2Rw}, \text{2Fw}, \text{2Bw} >$. This subgroup exists of all permutations p that can be solved with only moves from the set $\{\text{2Uw}, \text{2Dw}, \text{2Lw}, \text{2Rw}, \text{2Fw}, \text{2Bw}\}$. This group is also isomorphic to $G(3)$.

An interesting observation is that $|L| = |G(3)|^2$. Intuitively this makes sense, since the set L is constructed in a way that when the inner 3x3x3 representation of L is solved like a 3x3x3 Rubik's cube, but with wide moves, the result is an element of $G(5)_{\text{out}}$ which can also be solved like a 3x3x3 Rubik's cube. L is equal to the union of all cosets of $G(5)_{\text{in}}$ of elements in $G(5)_{\text{out}}$.

$L = \bigcup_{g \in G(5)_{\text{out}}} G(5)_{\text{in}} \cdot g$

The state of $G(5)_{\text{out}}$ is independent from the inner 3x3x3 representation of the centers, therefore the size of L is the size of $G(5)_{\text{out}}$ multiplied by $G(5)_{\text{in}}$, which is equal to $|G(3)|^2$.

7 Proposed algorithm

The algorithm exists of 4 steps, each step will reduce the size of the set of possible states the cube can be in, and the last step reduces the set of possible states to only the solved state I.

let $p \in G(5)$ be an arbitrary state of the 5x5x5 Rubik's cube. A sequence of moves to solve p, can be found in the following steps:

7.1 Step 1: Orienting the centers

Find $w = w_1 w_2 \dots w_n$ with $w_i \in \{\text{U}, \text{2U}, \text{D}, \text{2D}, \text{L}, \text{2L}, \text{R}, \text{2R}, \text{F}, \text{2F}, \text{B}, \text{2B}\}$ such that $(pw) \in K \subset G(5)$
This orients the centers in a way that they represent a 3x3x3 Rubik's cube.

7.2 Step 2: Solving the edges

Find $x = x_1 x_2 \dots x_n$ with $x_i \in \{\text{U}, \text{D}, \text{L}, \text{R}, \text{F}, \text{B}, \text{2U}, \text{E}, \text{2D}\}$ such that $(pwx) \in L \subset K \subset G(5)$
Solve the edges in a way that each pair of 3 adjacent edges would have the same colors and orientation if the 3x3x3 representation of the centers would have been solved.

7.3 Step 3: Inner 3x3 stage

Find $y = y_1y_2 \ldots y_n$ with $y_i \in \{2\text{Uw}, 2\text{Dw}, 2\text{Lw}, 2\text{Rw}, 2\text{Fw}, 2\text{Bw}\}$ such that $(pwxy) \in G(5)_\text{o}$ Solve the 3x3x3 representation of the centers using wide moves, that move both inner and outer layer of each face. This will result in a state for which all 3x3 centers are solved, and all adjacent edges have the same colors and orientation. Solving the 3x3x3 representation can be solved with any 3x3x3 Rubik's cube solving method. In this paper the most efficient method, Kociemba's Algorithm [9], is used. This always solves the 3x3x3 Representation in at most 20 moves.

7.4 Step 4: Outer 3x3 stage

Find $z = z_1z_2 \ldots z_n$ with $z_i \in \{\text{U}, \text{D}, \text{L}, \text{R}, \text{F}, \text{B}\}$ such that $(pwxyz) = I\}$. Solving a state of $G(5)_\text{o}$ can be done with any 3x3x3 solving algorithm, but just as in step 3, Kociemba's Algorithm is used [9].

A solution to solve state p is $wxyz$.

8 Correctness

8.1 Step 1 & 2: Orienting the centers and solving the edges

To determine whether it is possible to find a sequence of moves to to go from an arbitrary state to a state in the set K, first the parity of a state needs to be defined. Each move can be decomposed in cycles of individual pieces. This can be decomposed further to 2-cycles. The amount of 2-cycles is either odd or even, regardless of the decomposition. When there are an even amount of 2-cycles the parity is even, and for an odd amount of 2-cycles the parity is odd.

Each inner move consists of 4 4-cycles of the pieces of that slice, a 4-cycle of the edge pieces, and three 4-cycles of the centers pieces. Each 4-cycle can be written as 3 2-cycles. $(abcd) = (ad)(ac)(ab)$. Hence an inner move consists of 4 times 3 equals 12 2-cycles of the pieces. Each outer move of a face consists of 6 4-cycles of the pieces, a 4-cycle of the corner pieces, 3 4-cycles of the edge pieces, a 4-cycle of the 4 corner-center pieces of that face, and a 4-cycle of the edge-center pieces of that face. Hence an outer move consists of 6 times 3 equals 18 2-cycles of the pieces.

Each single move on the 5x5x5 Rubik's cube is either an inner or an outer move, each move can be decomposed in an even amount of 2-cycles, hence the parity remains the same after each move. The parity of the solved state, the identity permutation, is even. Therefore all permutations of reachable states from the solved state are even.

Theorem 8.1 *The parity of all reachable states of the 5x5x5 Rubik's cube are even.*

Since the parity of the all states is even, it is impossible to swap 2 pieces while leaving the rest of the cube unchanged. This would require 1 single 2-cycle, so the state must have odd parity, which can not exist. But it is possible to cycle 3 pieces from any orbit around while leaving the rest of the cube unchanged. This result can even be generalized further to the following theorem:

Theorem 8.2 (Composition of even permutations in 3-cycles) *Every even permutation can be decomposed in 3-cycles.*

To proof this theorem, consider the permutation:
$p = (a_1\ b_1)(a_2\ b_2)\ldots(a_{2n-1}\ b_{2n-1})(a_{2n}\ b_{2n})$
For each $(a_{2i-1}\ b_{2i-1})(a_{2i}\ b_{2i}) = (ab)(cd)$ there are 3 nontrivial cases:

- (a b)(c d), each element in these two 2-cycles are distinct, and their composition can be written as the composition of the following two 3-cycles: (d c a) (a b c)
- (a b)(a d), the composition can be written as a single 3-cycle: (a b d)
- (a b)(c b), the composition can be written as a single 3-cycle: (a c b)

For the trivial case (a b)(a b), the composition of these 2 cycles is the identity.

Each arbitrary pair of 2-cycles $(a_{2i-1}\ b_{2i-1})(a_{2i}\ b_{2i})$ is equal to the composition of 3-cycles. Therefore p must be equal to the composition of 3-cycles. Since p is chosen arbitrarily, the theorem holds for every even permutation.

In the context of the 5x5x5 Rubik's cube, this means that since every state is an even permutation, the difference between 2 arbitrary states A and B can be written as the sequence of moves from A to B, or an even amount of swaps of 2 pieces. This must be equal the composition of cycling 3 pieces. Take A and B such that only 3 pieces are changed while the rest of the cube is the same for both A and B. There must exist a composition of swapping 3 pieces to get from A to B, which is a single 3-cycle. But there also must be a sequence of moves from A to B, since both are reachable states. Hence there must exists a sequence of moves for any arbitrary 3-cycle. There must exists a sequence of moves that cycles 3 corner-center pieces, and a sequence of moves that cycles 3 edge-center pieces.

Notice that since the cube has equivalent corner-center and edge-center facelets, it is possible to seemingly swap 2 corner-centers or 2 edge-centers, using a 3 cycle with 2 pieces with equivalent facelets. For example, the sequence of moves: [2R' F' 2L F 2R F' 2L' F] U [2R' F' 2L F 2R F' 2L' F] U' swaps 3 corner-centers on the top and front face. The algorithm: [2R' F' M F 2R F' M' F] U [2R' F' M F 2R F' M' F] U' swaps 3 edge-centers on the top and front face.

There are 24 corner-center pieces, and 24 edge-center pieces. If for every 3-cycle only 1 piece is solved, and 3 for the last 3-cycle, there are 22 3-cycles needed for the corner-centers and 22 for the edge-centers, to solve them in the solved state of a 5x5x5 Rubik's cube. However, for the first step of the proposed heuristic, the centers do not need to be solved, but they need to represent a reachable 3x3x3 Rubik's cube state. This means that there are 43,252,003,274,489,856,000 possible states of the centers to move towards. When the closest state is chosen, a lot less 3-cycles have to be made on average and in the worst case.

The same argument can be made for solving the edges of the cube in a way that they form pairs of 3 similar edges when the 3x3x3 representation of the centers would have been solved. An element in L can be chosen that is closest to the current state. And the difference can be solved by 3-cycles of edge pieces. Without loss of generality, the middle edges can be considered to be solved at the start. Then the 2 outer edges corresponding to each middle each need to be solved in a place that would pair them up with the middle edge when the 3x3x3 representation of the centers will be solved. This means that only 24 edge pieces need to be solved in this step.

It is important to note that cycling pieces around is not an efficient way to solve a Rubik's cube, since it requires many moves for a few solved pieces. But it is not unreasonable to assume that this step could be improved by a lot in the future. Just like how reducing the maximum required amount of moves for methods to solve the 3x3x3 Rubik's cube took many years and many different people.

8.2 Step 3 & 4: Inner and outer 3x3 stage

By solving the edges in step 2, the state of the cube is reduced to an element of L. This set is equal to the union of all cosets of $G(5)_{\text{in}}$ of elements in $G(5)_{\text{out}}$.
$L = \bigcup_{g \in G(5)_{\text{out}}} G(5)_{\text{in}} \cdot g$

There is a bijective mapping between the states of $G(5)_{\text{in}}$ and $G(3)$. Every state is also a permutation, which can be factorized in individual moves. If $\phi : G(3) \rightarrow G(5)_{\text{in}}$ is the mapping between the two groups, and $p \in G(3)$ is permutation which can be factorized in moves from $\{\text{U}, \text{D}, \text{L}, \text{R}, \text{F}, \text{B}\}$ then $\phi(p) \in G(5)_{\text{in}}$ is a permutation which can be factorized in $\{2\text{Uw}, 2\text{Dw}, 2\text{Lw}, 2\text{Rw}, 2\text{Fw}, 2\text{Bw}\}$. It has been proven that every state in $G(3)$ can be solved with Kociemba's Algorithm [9]], this implies that every state in $G(5)_{\text{in}}$ can be solved as well. This means that every state in L is solvable to a state in $G(5)_{\text{out}}$.

There is a bijective mapping between the states of $G(5)_{\text{out}}$ and $G(3)$. Both use the same generator, and the permutation corresponding to the states can be factorized a sequence of moves from the same set. If $\psi : G(3) \rightarrow G(5)_{\text{out}}$ is the mapping between the two groups, and the factorization of p^{-1} is a solution of $p \in G(3)$, then the factorization of $\psi(p^{-1})$ is a solution $\psi(p) \in G(5)_{\text{out}}$. Which implies that every state of $G(5)_{\text{out}}$ is solvable with Kociemba's Algorithm.

9 Efficiency

To state something about the efficiency of the proposed algorithm, the move count should be analyzed. My initial approach was to model the 5x5x5 Rubik's cube, and to implement the algorithm in python. Then I would run a simulation of my algorithm, and an already existing, very efficient, algorithm, with the same cube states as input. However I did not manage to implement my algorithm on time, I do have a half working algorithm, which can be found on github [4]. I do have a working implementation of the 5x5x5 Rubik's cube, and I did manage to find an efficient 5x5x5 Rubik's cube solver. So I did generate 1000 cube states with my 5x5x5 Rubik's cube model, then I let the efficient algorithm solve them, and I put the solutions back in my own model to validate them. So I do not have an average move count of my own heuristic algorithm, but I do have an average move count for the already efficient algorithm, which gives an indication for how efficient the steps of my heuristic at least need to be for the heuristic to be an improvement compared to existing methods.

9.1 Efficient 5x5x5 Rubik's cube solving algorithm

I used the efficient algorithm from Daniel Walton [11]. This algorithm can solve cubes of any size, but it has specific optimizations for all N $\leq$ 7. It makes use of huge lookup tables with precalculated partial solutions to reduce the calculation time. The algorithm exist of 7 steps which are carefully chosen such that it is feasible to find the required moves to go from one step to another with a combination of brute force and lookup tables within a reasonable amount of time.

It is outside the scope of my project to explain how the efficient algorithm works in detail, but I will explain the 7 steps in short. In the first step the left and right center are solved in a way that all center pieces are of one of the 2 colors from the left and right side. So for example, if the left and right side on the solved cube are green and blue, then all center pieces of the left and right center will be solved in a way that they are either blue or green. In the second step, the same will be done for the 2 remaining centers. In this way all center facelets have either the solved color, or the color opposite of the solved color. In the third and fourth step, the edges are grouped in a way that they can be solved more efficiently. In the fifth step the 4 edges in between the L, R, B and F face are solved, and the L, R, B, and F centers are solved in a way that there are only vertical bars of the opposite color left unsolved. In the sixth step all edges and centers are completely solved. And the 7th step solves the remaining state completely as a 3x3x3 Rubik's cube.

The algorithm requires a string of 150 characters, representing the facelets of the cube, as input. And it gives a sequence of moves that would solve the given state as output. How the input is formatted is explained in more detail in the README.md file of my own implementation [4]. Since the same input format is used.

9.2 5x5x5 Rubik's cube model in python

I made an implementation of the described model of the 5x5x5 Rubik's cube. When initializing a cube, it takes a string of 150 characters as input. Any move of any face in any direction for both the inner and outer layers has been implemented. And a function to randomly scramble the cube for an N amount of moves. Each state of the cube is modeled both as a list containing 6 lists with 25 facelets, one list for each face, and as a string of 150 characters. Where each character is either "U", "R", "F", "D", "L" or "B", representing the names of the faces they belong to in the solved state. The source code of the model can be found on github [4].

9.3 My attempt at implementing my proposed algorithm

I have tried to implement the proposed algorithm in python, but I did not manage to implement this in time. The main setbacks were that I did not find existing, usable, code for the second step of my algorithm, for solving the edges. This did mean that I had to implement this myself as well as the first step of the algorithm. Since the third and fourth step are the same as solving a 3x3x3 Rubik's cube, these steps are not a problem to implement. The kociemba package [8] for python can be used to solve a 3x3x3 Rubik's cube. This uses the same input format, and returns a string of moves.

Another problem I ran into was that I was not able to find a generalized algorithm to cycle 3 arbitrary pieces around. This is always possible, which has been proven in the section about correctness, but I could not find a general approach to find the sequence of moves needed to cycle these 3 arbitrary pieces around. While it can be relatively simple to do it yourself on a 5x5x5 Rubik's cube, it can be hard to formalize the logic behind what you are doing to be able to implement it.

The most important realization was that my initial approach of finding the suitable 3 cycles did not work. I tried to match pieces to represent a corner, without taking the orientation into account. When a corner exist of 3 distinct symbols, there are 3! different configurations of those symbols, but only 3 of them represent a valid corner. For example if UFL represents the Up-Front-Left corner, then UFL, FLU, and LUF are valid corners. But FUL, ULF, LFU do not represent valid corners. I did not take this into account in my initial two approaches. In my third approach I did take this into account, but I still discovered some inexplicable cases, which is why the program sometimes gives an error while solving the corners of step 1. So I do have a partial implementation of the first step, which works most of the time, but not always.

9.4 Simulation

The model of the 5x5x5 Rubik's cube was used to generate 1000 strings of states which were all written in a file. This file can be found on github [4]. These strings can be used as input for the efficient algorithm [11].

The efficient algorithm is built for linux, therefore the simulation has been performed in the WSL environment. This did have some advantages. The efficient algorithm has many print statements for extra information about intermediate steps during the solving of a single state. This is rather inconvenient if you want to run the program 1000 times. But every print statement is seen as error handling. While the actual solution is only seen as output. Therefore the program is runned, in bash for example, followed by the command "2> /dev/null 1» output.txt", it will write all unnecessary intermediate prints in /dev/null, and it will append the actual solution to output.txt.

I made a file that ran the efficient algorithm 1000 times, with the 1000 different states as input which appends all solutions to output.txt. But the first time I ran the simulation, I discovered that I only got 999 results. So when I ran it again, but with numbered solutions, I saw that solution 577 was missing. When I manually ran it I discovered that the efficient algorithm gave an error for this state. This means one of two things, either my 5x5x5 Rubik's cube model is wrong, and can generate impossible input states, or there is a bug in the efficient algorithm. I managed to find a solution for this state, which will be explained in more detail in the next section. This implies that the efficient algorithm got 1 of the 1000 states incorrect. I got the following results from the simulation:

	Correct	Incorrect
Amount	999	1
Avg move count	79.13	-

I also tested my partially working implementation of the first step on the same 1000 states as for the simulation of the efficient algorithm. The detailed results can be found in 'simulations_results.txt'. The simulation counts the amount of 3-cyles needed for the first step. The following frequencies of the results were found:

Amount of 3-cycles	Frequency
2	79
3	244
4	154
5	35
6	82
7	68
8	38
9	15
10	10
11	11
12	9
13	8
14	3
18	2
20	4
21	4
23	1
25	1
26	1
Error	221

9.5 Validation

To validate the results, I ran all solutions that the efficient algorithm returned in my own 5x5x5 Rubik's cube model. All 999 solutions are correct. To confirm whether the error for one of the input states was a bug from the efficient algorithm or the 5x5x5 Rubik's cube model, I tried to solve it with another solver. The state input is: BBRLRBUL-BUDRUFRFLLLLDDFLL BUFRUFDDFRDRRBLDFLFFDDBDD RRUDUURFRRBL-FUFBLBDRBRRLF RFUBLFDRDBDUDDUFLBBLFBLRR LULUBFUDULLULBDBR-RBRUDRLU FFBLDUFUUDBDBFFURFBUFBUDL. I used the online solver of Grubiks [3]. This gave a solution: 2Rw F' 2Uw L B2 2Dw 2Lw' F' 2Lw' D' 2Bw R' 2Dw2 L' U 2Bw' 2Dw2 L' 2Fw' 2Uw2 B' D' R U 2Uw2 D2 F' D 2Dw2 F2 2Lw2 U2 2Lw2 U' R2 2Uw2 2Lw2 2Fw2 2Lw2 B2 U 2Dw2 L 2Fw2 2Dw2 R2 2Fw2 R2 D2 R 2Bw2 U2 2Fw2 U2 L 2Bw2 B' U2 B' U F U B2 R' U L B' U2 L B2 R' D2 L' D2 F2. I used this solution in my 5x5x5 Rubik's cube model, and it did indeed solve the input state, which means that the state was not incorrect.

10 Conclusions & Future work

In this paper I did try to introduce a new heuristic algorithm to solve the 5x5x5 Rubik's cube. I also did not manage to calculate the average move count of the proposed heuristic. But I did find the average move count of an efficient algorithm. The efficient algorithm solved the input states in 79.13 moves on average. Since steps 3 and 4 of the proposed heuristic are equivalent to solving a 3x3x3 Rubik's cube. it is known that this will never take more than 20 moves, and roughly 19 moves on average [9]. This leaves roughly 41 moves on average left for the first 2 steps of the heuristic to become as fast as the efficient algorithm. It is uncertain whether this could be possible, since the subsets K and L are newly defined sets and not much is known about them. My research question was:

In what way can a 5x5x5 Rubik's cube be efficiently solved from an arbitrary state?

An answer to this question is that the 5x5x5 Rubik's cube can be solved efficiently using the efficient algorithm which has been discussed in section 9. This is a rather unsatisfying answer, since the aim was to show that my own algorithm was a way to efficiently solve the 5x5x5 Rubik's cube from any state. But I did not manage to show or prove that my algorithm can be efficient. However, I can not even claim that the efficient algorithm is a way to efficiently solve the 5x5x5 Rubik's cube from an arbitrary state, since I had found a state for which it was not able to solve it. This means that I do not have an answer to my initial research question. But I do have new questions, and new recommendations for future work, since the research of solving the 5x5x5 Rubik's cube efficiently is far from finished.
The problem of solving a state of $G(5)$ to an state of K is seemingly similar to solving a 3x3x3 Rubik's cube, with the important difference that there is not 1 but $1.16 \cdot 10^{63}$ solved states, and solved states aren't easily recognisable by colors. These differences make it really hard for humans to get the right intuition for it, but for computers it shouldn't make a difference. Therefore it is not unreasonable to assume that the solving of the first state can be optimized a lot when the right method for it is found. Trying to optimize the first step, and finishing the implementation of the heuristic to get the average move count by simulation are both obvious challenges left for future work. A potential question for further research is:

What is the minimum required number of moves to solve an arbitrary state of $G(5)$ to an element of K?

This is somewhat similar to finding the diameter of the 3x3x3 Rubik's cube [9]. Another important thing to mention is that more peer reviewed research about the 5x5x5 Rubik's cube, or Rubik's cubes in general, can be done. I could not find a published, and peer reviewed paper about the size of the NxNxN Rubik's cube permutation group. This is also something that should be looked at in the future. But the most important work left is analyzing and optimizing the proposed heuristic, to investigate whether this is an improvement on existing algorithms.

References

[1] Erik D. Demaine, Sarah Eisenstat, and Mikhail Rudoy. Solving the rubik's cube optimally is np-complete. *CoRR*, abs/1706.06708, 2017.

[2] K Fraser. Rubik's cube extended: Derivation of number of states for cubes of any size and values for up to size 25x25x25.

[3] Grubiks. Professor's cube solver 5x5x5. `https://www.grubiks.com/solvers/professors-cube-5x5x5/`.

[4] Xander Heij. Model of the 5x5x5 rubiks cube in python. `https://github.com/s2571250/5x5x5RubiksCubeProject`, 2023.

[5] Ahmad Kaleem and Ahsan Kaleem. On algorithms for solving the rubik's cube. *CoRR*, abs/2007.10829, 2020.

[6] Oliver Knill and Roman Maeder. The rotation group of rubik's cube. *SIGSAM Bull.*, 21(3):33–43, 1987.

[7] Prajna Kunche, KVVS Reddy, Prajna Kunche, and KVVS Reddy. Heuristic and metaheuristic optimization. *Metaheuristic Applications to Speech Enhancement*, pages 17–24, 2016.

[8] Muodov. kociemba 1.2.1. `https://pypi.org/project/kociemba/`.

[9] Tomas Rokicki, Herbert Kociemba, Morley Davidson, and John Dethridge. The diameter of the rubik's cube group is twenty. *SIAM Rev.*, 56(4):645–670, 2014.

[10] David Singmaster. *Notes on Rubik's magic cube.* Enslow Pub Incorporated, 1981.

[11] Daniel Walton. rubiks-cube-NxNxN-solver, April 2017.

Notation List

Gen($G(5)$) The set $\{\mathrm{U}, 2\mathrm{U}, \mathrm{D}, 2\mathrm{D}, \mathrm{L}, 2\mathrm{L}, \mathrm{R}, 2\mathrm{R}, \mathrm{F}, 2\mathrm{F}, \mathrm{B}, 2\mathrm{B}\}$. (3.2)

Algorithm Sequence of moves, performed from left to right. (2.3)

Big-cubes NxNxN Rubik's cube larger than 3x3x3. (2.1)

Center-center The middle of the 3x3 center. (2.1)

Corner-center A corner center piece. (2.1)

Cubie One of the smaller individual cubes that forms the Rubik's Cube. (2.1)

Edge-center A edge center piece, in between two corner-centers. (2.1)

Equivalent States states that look the same on a physical Rubik's cube. (3.3)

Equivalent states Two states are equivalent when all facelets are equivalent. This means that you can not see a difference between them on a physical cube. (3.3)

Face Side of a Rubik's cube, a cube has 6 different faces. (2.1)

Facelet Visible face of a cubie. (2.1)

Half Turn Metric (HTM) The metric for which a turn of a single face counts as a single move. (3.4)

Move Turn of a single face, either 0, 90, 180, or 270 degrees. (2.2)

Parity of a permutation Odd when the permutation can be decomposed in an odd amount of 2-cycles, and even when the permutation can be decomposed in an even amount of 2-cycles. (8.1)

Piece Cubie of a physical Rubik's cube, either a center piece, edge piece or corner piece. (2.1)

Size of permutation The amount of moves, in STM, needed to solve the state corresponding to the permutation (3.5)

Slice Turn Metric (STM) The metric for which a single synchronous turn of adjacent slices in the same direction on the same axis counts as a single move. (3.4)

Solved state The state for which all faces of the cubies on each face have the same color. This is also the factory state of a physical Rubik's cube. (2.1)

The group G(5) The permutation group of all reachable states of the 5x5x5 Rubik's cube. (3.2)

The set S The set of all facelets. (3.1)