# CAB203 Project – 2 x 2 Rubik's Cube Report

Alexander Kim

n10794565

## 1 Introduction

The standard and the most famous, 3 x 3 Rubik's cube has a smaller version that comes in a 2 x 2 cube. The smaller 2 x 2 Rubik's cube inherits most of the features of its bigger brother having the same number of faces and colours just with fewer squares per face. The 2 x 2 cube's 6 faces are broken down into four squares in each face. In the solved state, the squares on the faces are all coloured orange, green, red, blue, white, and yellow as shown below. Compared to the standard 3 x 3 Rubik's cube which allows for the faces to be manipulated around the core, the 2 x 2 cube faces are manipulated by twisting one of its faces. Therefore, twisting the right face clockwise is the same and twisting the left side anticlockwise.

The purpose of this puzzle toy is to scramble the cube by twisting the faces randomly for a certain number of times to randomise the colours of the squares to initialise the cube to then be solved back into its original state. The cube is then solved through manipulating the cube through moves. A single move in this case is considered to be a single 90° turn in either anti-clockwise or clockwise direction for any of the faces. The cube is deemed solved if the colours of the 4 squares on each face are all matched with the same colour. This project will aim to find the smallest number of moves needed for a given scrambled cube to be solved.

## 2 Instance model

The act of manipulating the cube by performing a move to arrive at a certain configuration can be described by a set of configuration-to-configuration links where a single move is connected between the configurations. Using this information, the state of configuration of the cube, $u$ and $v$ (where $u$ is one configuration and $v$ is the next configuration) are an edge $(u, v)$ where a valid move exists so that from $u$, it is possible to proceed to $v$ by a single move.

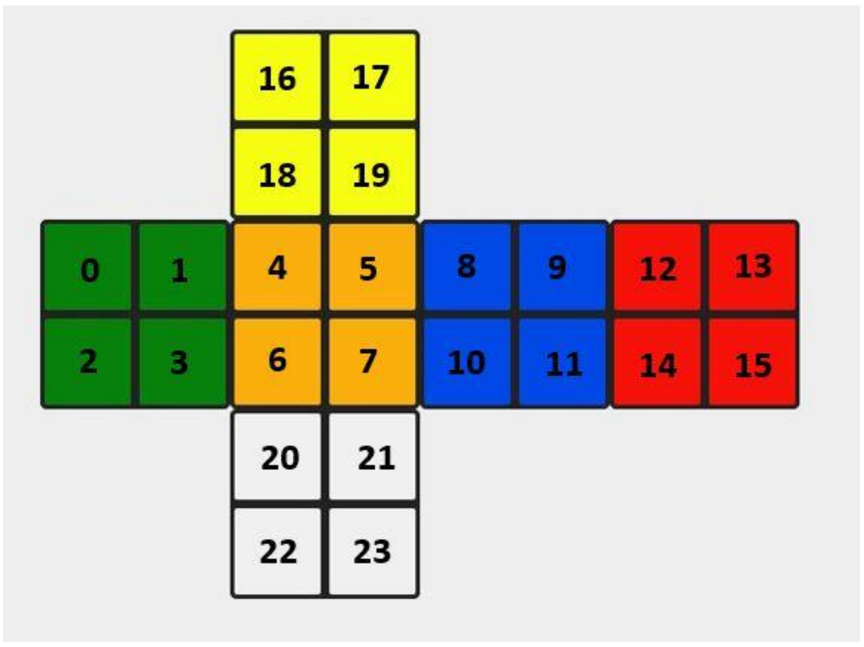
However, there are 6 different moves, front clockwise, front anticlockwise, up clockwise, up anticlockwise, right clockwise and right anticlockwise that can be manipulated to $u$ to result in different $v$. Therefore, the number of vertices that are in an edge from the previous $u$, will exponentially increase up to a certain point with each unique configuration then decrease back down as there are no newer unique configurations. Rubik's cube enthusiasts will go by something known as the "God's number" which is the maximum number of moves required to solve any cube. In this case, it is 14 considering quarter turns to be an only move. In other words, the degree of the vertex is 14. Due to the nature of the problem, it is impossible to model all 3, 674, 160 permutations of vertexes (2015, para. 7). However, a generalised set for a single move can be defined:

$$M = \{(u, v)\}$$

An instance model can be formed by $(M, s, f)$ where $M$ is a set of vertices (configurations), $s$ is the starting configuration and $f$ is the final solved configuration as shown below:

$$(\{(u, v) \ldots\}, Starting\ config., Final\ config.)$$

The instance of the problem is given by a random scrambled configuration of the cube. The cube in this report and the python code will encode the cube from numbers 0 to 23 as illustrated below:



This solved configuration of the cube will encode like so:

GGGGOOOOBBBBRRRRYYYYWWWW

## 3 Solution model

To find the shortest number of moves to solve the cube from a given starting configuration, the cube must be manipulated by a certain number of moves to arrive at a solved configuration. Hence the solution to the problem is to count the number of moves taken from the starting configuration to the solved state. The solution model to the problem will just be an integer.

## 4 Problem model

The model to the problem can be modelled by a graph, $G = (V\ , E)$. The graph will consist of $V$ which are the possible configurations of the cube while $E$ consists of an edge for each sequence of configurations accessible by a single move.

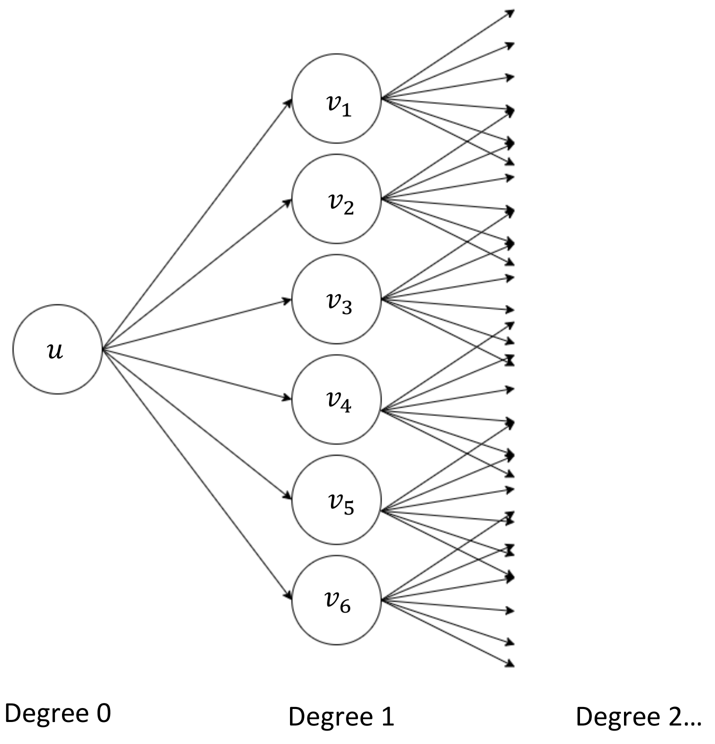
The set of vertices, $V$ is given by:

$$V = \{u : (u, v) \in M\} \cup \{v : (u, v) \in M\}$$

The set of edges, $E$ is given by:

$$E = \{(u, v) : (u, v) \in M\} \cup \{(v, u) : (u, v) \in M\} \cup$$

A visual representation of the graph, $G$ can be modelled for the first 2 depths of the problem where the number of vertices and edges are manageable. Each degree from the vertices to the next represents an edge where a valid move exists. The following diagram is a visual representation of $G$.



Each of the 6 vertices represented by circles in Degree 1 represents a unique configuration from the starting configuration in Degree 0 from applying the unique 6 possible moves. From each of the vertices in Degree 1, another set of 6 moves can be applied to obtain new set of vertices. Any overlapping configurations (vertices) will be linked together.

The solution of the problem can be found by starting from $s$, going through each of the vertices, $V$ at every degree to check if the move has resulted in a solved configuration, $f$. The degree travelled from the starting point will be counted until it has reached $f$. Hence, the solution will be the shortest number of degrees evaluated from $s$ to $f$.

## 5 Solution method

To find the shortest number of degrees, breadth first traversal will be utilised. The solution will essentially feature a loop that checks for every vertex in every degree to see if the move has in fact solved the cube. The number of degrees needed to compute for this to happen is the minimum number of moves needed for any given scrambled 2 x 2 cube to be solved.

In order to implement a breadth first traversal algorithm, a distance (degree) class needs to be defined which are the set of $D_j$. This distance class, $D_j$ will start from $D_0$ with just the original vertex $u$ all the way to $D_j$. Each successive set of $D_j$ will be a new set of $V$ containing a new unique set of $V_j$ with the moves applied from the previous set of $V$. The set $D$ can be written as:

$$D = \left(\{D_0 \in V_0\}, \dots \{D_{j-1} \in V_{j-1}\}, \{D_j\}\right)$$

To compute the successive sets of $V_j$ a function to calculate the neighbourhoods from $V_{j-1}$ needs to be defined. Since there are 6 unique valid moves for a 2 x 2 cube, 6 functions can be hardcoded to simulate the order of the encoded coloured squares with its respective move applied. The 6 functions of moves can be arranged in a set:

$$moves = \{F, Fc, R, Rc, U, Uc\}$$

Where another function to process $V_j$ with the set of $moves$ to $V_{j-1}$ can be devised:

$$N_{moves}(v) = \{v \colon \{moves\}\}$$

At step 0 of $D$, $D_0$ is simply $u$ where it is the starting scrambled configuration. The distance at $D_0$ is 0:

$$D_0 = \{u\}$$

The next case can be calculated with $N_{moves}(v)$ applied to the previous vertices or in this case just $u$ of $D_0$, to calculate $V_1$:

$$V_1 = \{N_{moves}(V_0)\} = \{\, N_{moves}(u)\}$$

For every vertex in $V_1$ resulted from the $N_{moves}(v)$ function applied to the previous set of vertices, $V_0$ in $D_0$, the function $N_{moves}(u)$ is applied again to obtain $V_{j+1}$ which in this step is $V_2$. Every vertex of $V_2$ can be checked to see if the solved configuration of the cube is found. This is achieved through a predefined function which returns true if any of the vertices in $V_2$ is matched with any possible solved solution. Since the orientation of solved colours of the cube's faces can vary, the possible solution was formed using regular expressions that allows any block of 4 letter combinations representing of colours to be mixed. If true, the loop is broken, and the current set of $D$ index subtracted by 1 to account for $D_0 = 0$ is returned and printed to the user.

Otherwise, the vertices in $V_2$ can be filtered to create a new unique set of vertices, $V_j$ to be appended to the next $D_j$ which in this case is the corresponding $D_2$. A property of the cube is realised where performing a single move on a cube can only scramble the cube either one degree more $(D_{j+1})$ or one degree less $\left(D_{j-1}\right)$. Implementing this property will allow the new neighbouring vertices derived from the previous set of vertices to be a unique set where any repeating vertices are omitted as that essentially means that the move applied to scramble has solved the cube by one move instead. This can be processed by the following:

$$V_j = V_j \setminus V_{j-1} \cup V_{j-2}$$

The new set of $V_j$ is the set of vertices for the respective distance, $D_j$. The algorithm continues finding the next set of $V$ for the next respective distance then checks to see if the solution is found within one of the vertices or continues filtering for new vertices then proceeds to the next distance until there are no unique number of vertices. Since the maximum degree of a 2 x 2 Rubik's cube is 14, all solutions will either be or under 14 moves. It will never exceed 14 moves as anything beyond 14 essentially means the moves are in fact unscrambling the cube where the previous same vertex has already been found in previous distances which is why the repeated vertices were omitted.

In summary, the solution can be found from an instance as follows:

1. Compute a set of $D$ with $D_0$ being just $u$, the starting scrambled configuration then calculate the next $D_1$ and its set of neighbouring vertices, $V_1$.
2. Calculate the neighbours of $V_1$ and check if a solution exists. Otherwise append the set of unique neighbours calculated from $V_1$ as $V_2$ in the next distance.
3. Calculate the neighbours of the last set of vertices in the latest distance, $D_j$ and check for any solutions. Otherwise append the set of unique neighbours from $V_{j-1}$ as $V_j$ in the next distance $D_j$.
4. Repeat step 3 until the solution is found.

## 6 References

Scherphuis, Jaap.(2015). *Mini Cube, the 2×2×2 Rubik's Cube.* Jaap's Puzzle Page. https://www.jaapsch.net/puzzles/cube2.htm