Rubik’s Cube in 42 Moves

By Hans Kloosterman

1 INTRODUCTION

Last year I have worked on a short algorithm to solve the Rubik’s cube. Now it has reached it’s final length of 42 moves.

Before explaining the algorithm in more detail I will give a short overview of the algorithm. The four stages of the algorithm are:

1. Within the group <R,L,F,B,U,D> reduce it to <R,L,F^2,B^2,U,D>	(max 7 moves)
2. Within the group <R,L,F^2,B^2,U,D> reduce it to <R^2,L^2,F^2,B^2,U,D>	(max 10 moves)
3. Within the group <R^2,L^2,F^2,B^2,U,D>, put all U-cubies in the U-face (implicitly all D-cubies are in the D-face) 4. Within the group <R^2,L^2,F^2,B^2,U,D> restore the cube	} (max 25 moves)

The separate numbers of moves for the stages 3 and 4 are respectively 8 and 18. The total number of moves for these stages becomes 25 because in the situations of 18 moves ion stage 4 the first move of these processes is cancelled against the last move of stage 3.

The numbers of situations for each stage are: 2048, 1082565, 4900 and 3981310.

2 THE METHOD

The numbers of processes I have generated to solve the cube according to this algorithm are at this moment respectivily 2048, 25534, 1276 and 30784. It is possible these numbers for stages 2 and 4 will change a bit during the reorganisations of the tables. In stage 4 there are 6 situations out of 30784 which are 18 moves long. Although I have not proved it yet completely, it seems very unlikely to me these situations are possible in 17 moves.

2.1 Stage 1

The purpose of this stage is to orient all edges. Because it is difficult to understand what happens I will explain this. Imagine a solved cube and colour it with two colors like fig. 1.

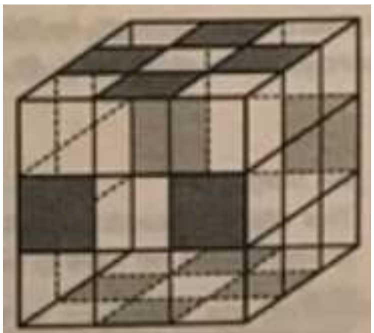
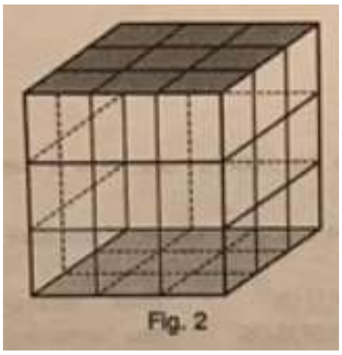


Fig. 1

On this cube all moves are allowed. The purpose of this stage can be stated as: restore a cube coloured this way.

Hence <R,L,F^2,B^2,U,D> leaves this invariant, so stages 2 will not destroy the result of stage 1. Another property of this stage is that when the process has at least one move it ends with a quarter F or B move. In case the last move is a quarter F move it doesn’t matter whether it is an F or F’. Analogically there is no difference between B and B’.

2.2 Stage 2



The purpose of this stage is to orient all corners and put the edges from the U-D slice into this slice. It is possible to explain this with a two coloured cube.

Let's consider a clean cube and colour this like fig. 2.

The U and D faces are coloured black and the other faces white. On this cube only the moves out of the group $<R,L,F^2,B^2,U,D>$ are allowed. Hence these moves don't change the orientation of the edges. The purpose of this stage is to restore this cube.

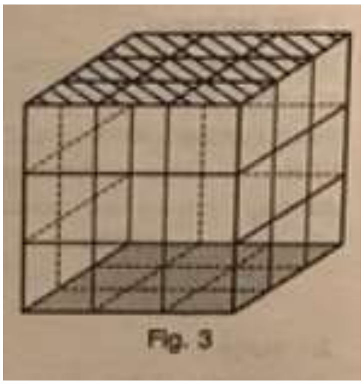
The last move of this stage is R, R', L or L'.

The result of stage 1 and 2 is that all edges and corners are oriented correctly and the edges of the U-D slice are in this slice.

Using the moves of the group $<R^2,L^2,F^2,B^2,U,D>$ leaves stages 1 and 2 invariant. SO by using these moves for stages 3 and 4 we don't need to worry about disturbing stages 1 and 2.

2.3 Stage 3

In this stage all the cubies of the U-face are put in the U-face. Because all the cubies of the U-D-slice remain in their slice (stage 2) all the cubies of the D-face are in the D-face.



It is also possible to explain this with a cube with three colours. The U and D face are coloured each with a different colour. All other faces have the same colour. On this cube only moves from the group $<R^2,L^2,F^2,B^2,U,D>$ are performed. The purpose of this stage can be seen as to restore this cube.

The maximum length for this stage is 8 moves. The last move of this stage is R^2, B^2, L^2 or F^2.

2.4 Stage 4

This stage is easy to describe: solve a normal coloured cube which is invariant to stage 3. This means we have to restore the permutation of the U-face, the D-face, and the U-D slice.

The maximum number of moves is 18. For these situations I have found alternatives which make it possible to starts with each of the moves R^2, B^2, L^2 and F^2. Because the processes with length 8 or stage 3 always end with one of these moves it becomes possible to cancel the last move against the first move of stage 4.

3 THE SEARCHING PROGRAMS

For the searching I used a PC (10MHz 80286). The programs are written in Turbo-Pascal with a few machine language routines.

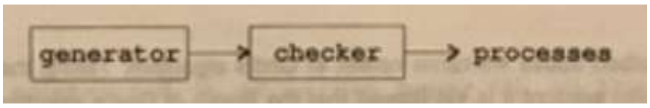
With the programs developed today it is possible to compute all processes in less than a week computing time. The first programs were not that fast. During the improvement of the algorithm these became a need for more efficient programs.

Another problem with the improvement of the algorithm was that I needed to use more situations for each stage. For instance with the reduction of stage 4 from 19 to 17 moves: in the case of 19 moves there were 4950 situations. I thought 27360 situations would be enough to reduce it to 17 moves. Unfortunately there were a few thousands exceptions, which leads to 30784 situations at this moment. Within these situations there were still 6 which needed 18 moves. However by using alternative processes it became possible to start with the moves R2, B2, L2 or F2. Because stage 3 always ends with one of these moves (in the case it has not length 0), it is possible to cancel the first move of stage 4 to the last move of stage 3.

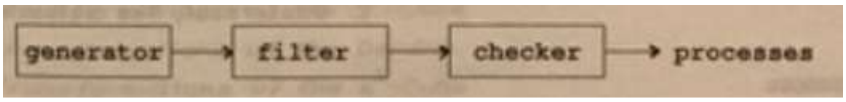
A major disadvantage of the reduction of stage 4 was that it became necessary to rewrite large parts of the programs due to the new organisation.

Another problem was to keep track to the number of situations in each stage, especially in stages 2 and 4. Reduction of situations is possible by eliminating rotations like U_C and R^2_C and reflection. It is also possible to introduce correction moves. This are intial or terminal U or D moves to set the U and D faces in a specific position.

To find the processes I used a generator which generates all possible processes within a group. This generator starts with a clean cube, so the generated processes are the inverses of the solving processes. In stage 1, 2 and 3 all possible processes have to be checked whether they are improvements



Due to the choice of the groups of allowed moves for the stages 1, 2 and 3 all generated processes in these stages belong to the group of solving processes, so they all have to be checked. IN stage 4 however not all generated processes are possible solutions. I used a filter to solve this problem. Only processes which don’t change the colour of the U-face are passed.



Generating all proesses has one disadvantage, a lot of proceses are generated several times. This is partly due to symmetries. I will now discuss how the number of duplicated processes can be reduced in stage 4.

The first move has only one possibility (R^2). The next possible moves are B^2, L^2, U and U^2. The third move has also limited possibilites.

For the second move F^2, U^2 and D are not produced because they lead to processes which are mirrors to other processes.

Another trick is to use after an L^2 never an R^2 move, and after a B^2 never an F^2move. Now sequences like $R^2L^2R^2$ are not produced.

When a process is an improvement, also the inverses and mirror processes are checked.

4 POSSIBLE IMPROVEMENTS

It is not possible to improve each of the separate stages within the allowed set of moves. However it may be possible to reduce the number of moves for the algorithm.

1. For every algorithm which uses several stages it is always possible to combine two or more stages into one. Mostly this will lead to a reduction of the length of the algorithm but also to a growth of the number of situations. It would not be easy to use this to reduce the algorithm according to this method because the number of situations would grow to much.

2. Another possible improvement is to allow in every stage all possible moves. In stage 1 it is not possible because all moves are allowed. In stages 2 and 3 I think it doesn't lead to an improvement.

In stage 4 it may be possible to reduce the number of moves. It is possible to give a lower bound for the length: the upper table gives solutions for a few cases of stage 4. These situations have a maximum length 15, so we may conclude that stage 4 requires at least 15 moves.

To reduce the number of moves for stage 4 the length has at least to be reduced to 16. I'm not convinced the use of quarter moves in stage 4 leads to such an improvement of the algorithm.

With the algorithm described above we came nearer to God's algorithm. Taking the results of the uppertable into account it is yet proved that the length of God's algorithm is between 16 and 42 moves.

There is still a pretty large gap between the upper and the lower boundary. This leaves an attractive goal for the next decade…

Hans Kloosterman

Oosterhout, The Netherlands

References

[1] Hans Kloosterman "Rubik's cube in 44 moves"

CFF 22 (December 1989) pp 9-11

[2] Hans Kloosterman "Rubik's cube not yet in 42 moves"

CFF 23 (march 1990) pp 10-11