

2006

Adventures with Rubik's UFO

Bill Higgins

Follow this and additional works at: http://ecommons.udayton.edu/mth_epumd



Part of the [Mathematics Commons](#)

eCommons Citation

Higgins, Bill, "Adventures with Rubik's UFO" (2006). *Undergraduate Mathematics Day, Electronic Proceedings*. Paper 14.
http://ecommons.udayton.edu/mth_epumd/14

This Article is brought to you for free and open access by the Department of Mathematics at eCommons. It has been accepted for inclusion in Undergraduate Mathematics Day, Electronic Proceedings by an authorized administrator of eCommons. For more information, please contact frice1@udayton.edu, mschlengen1@udayton.edu.

Adventures with
Rubik's UFO

Bill Higgins
Wittenberg University

Introduction

Erno Rubik invented the puzzle which is now known as Rubik's Cube in the 1970's. More than 100 million cubes have been sold worldwide. The mathematics behind solutions to the cube have been extensively studied by mathematicians and puzzle enthusiasts alike. See the book by Joyner [1] for example. Though I like puzzles, my history with the cube was one of failure. I found it too difficult for me to solve a mixed-up puzzle and memorizing a long complicated algorithm to solve the cube was not appealing to me. My youngest son became intrigued with the cube and asked for one of his own. He played for a while with a cube I gave to him and, like father - like son, discovered that the puzzle was too tough for him and asked me to solve the puzzle!

I decided to see what tools for restoring the puzzle were available on the internet and found that there are many sites devoted to online animated versions of the cube and automated cube solvers. Several such programs allow the user to input the initial mixed-up state of the cube and then provide a sequence of cube twists to restore the cube to the solved state. The program we used is called Cube Explorer [2]. A few months later, I gave the puzzle known as Rubik's UFO to my son for Christmas. Though he played with the puzzle longer than he had played with the Rubik's Cube, the result was the same. He returned the mixed-up puzzle to me to solve.

Solving Rubik's UFO

Though I suspected that an internet search for a solution to Rubik's UFO would be successful, this time I did not want to take that route. I wanted discover a procedure for solving the puzzle on my own. The first challenge was the fact that I wasn't even certain what the solved state of the puzzle looked like. In the holiday crush, the package the puzzle came in and the "hints booklet" could not be found. Here is a picture of a mixed up UFO puzzle.



Twelve pieces are arranged in two "hexagons" forming the top and bottom of the UFO puzzle and may be scrambled by using twists and rotations. In the solved state, the six grey pieces are on the top and six dark pieces are on the bottom. The colored markings on adjacent pieces also match as shown in a solved puzzle below. The bumps on three of the bottom pieces form the landing gear.

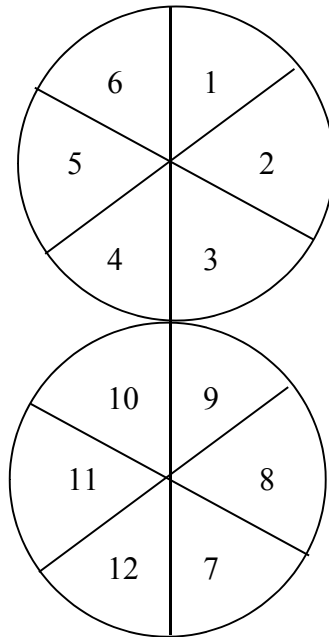


(puzzle in the midst of a twist)

A scrambling of the puzzle corresponds to a permutation of the puzzle pieces. My first step in solving the puzzle was to name the pieces. In the solved state, I chose the top piece containing the trademark label to be piece number 1 and continued naming clockwise on the top 2, 3, 4, 5

and 6. Naming the bottom pieces is done by adding 6 to each of the names of the top pieces so that piece 7 is directly below piece 1, piece 8 is directly below piece 2 and so on.

If the top and bottom of the puzzle were hinged like a pocket watch, then opening it to lay flat would produce the following diagram.



With this naming convention, a puzzle state may be represented by a 2 by 6 matrix whose entries contain the names of the pieces. Four basic manipulations of the UFO are described and illustrated below. A rotation of the top of the puzzle (denoted r below) or vertical twist of the puzzle would then correspond to a permutation of the entries of this matrix.

Initial position		After manipulation
$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$	60 degree clockwise rotation of the top $\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow r \rightarrow\rightarrow\rightarrow\rightarrow\rightarrow$	$\begin{bmatrix} 6 & 1 & 2 & 3 & 4 & 5 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$
$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$	twist the puzzle between pieces 1 and 6 $\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow v \rightarrow\rightarrow\rightarrow\rightarrow\rightarrow$	$\begin{bmatrix} 9 & 8 & 7 & 4 & 5 & 6 \\ 3 & 2 & 1 & 10 & 11 & 12 \end{bmatrix}$
$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$	twist the puzzle between pieces 1 and 2 $\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow x \rightarrow\rightarrow\rightarrow\rightarrow\rightarrow$	$\begin{bmatrix} 1 & 10 & 9 & 8 & 5 & 6 \\ 7 & 4 & 3 & 2 & 11 & 12 \end{bmatrix}$
$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$	twist the puzzle between pieces 2 and 3 $\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow t \rightarrow\rightarrow\rightarrow\rightarrow\rightarrow$	$\begin{bmatrix} 1 & 2 & 11 & 10 & 9 & 6 \\ 7 & 8 & 5 & 4 & 3 & 12 \end{bmatrix}$

In playing with the puzzle while attempting to solve it, I would write down the current state of

the puzzle then perform a sequence of rotations and twists (and write down what I did) hoping to make progress on a solution. In the course of such play, I discovered that the following sequence of moves, which I will call m , would switch two adjacent pieces on the top and would also switch the two pieces directly below. To apply m to the puzzle, first apply the vertical twist, v , then a 60 degree clockwise rotation, r , then the twist x , then a 120 degree counter clockwise rotation, r^4 , and so on as indicated below.

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix} \xrightarrow{\text{apply } m = v \cdot r \cdot v \cdot r^4 \cdot x \cdot r \cdot v} \begin{bmatrix} 1 & 2 & 4 & 3 & 5 & 6 \\ 7 & 8 & 10 & 9 & 11 & 12 \end{bmatrix}$$

The first step in solving a mixed-up puzzle is to move all of the light pieces to the top and the dark to the bottom. (This can be done fairly easily.) Then using m and appropriate rotations repeatedly it is possible to put the top half of the puzzle in the correct order. One can then turn the puzzle over and put the bottom half in order using the same method. As this is done, pieces 3 and 4 will swap positions with each application of m . If you use an even number of applications of m to put the bottom half in order, the puzzle will be solved. If not, it will look like this:

$$\begin{bmatrix} 1 & 2 & 4 & 3 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$$

How do we switch the 4 and 3? If we just apply m once more, the puzzle will look like:

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 10 & 9 & 11 & 12 \end{bmatrix}$$

so that the top looks fine, but the bottom is now out of order.

Rather than just applying m , what is needed is a sequence of moves in which m is involved and applied an *even* number of times. That way each switch of the 9 and 10 in the bottom row is reversed by a subsequent application of m .

Consider again our unlucky position below.

$$\begin{bmatrix} 1 & 2 & 4 & 3 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$$

When we switch the 4 and 3, via an application of m , it is as if we are moving the 4 past the 3 to put the top row in order. But if we move the 4 in the opposite direction - past the 2 then past 1 then past 6 and finally past 5, each via an application of m , we will have an *even* number of applications of m and the bottom row will be in order! Of course, to actually use m to switch the 4 and 2 we need to first rotate the top using r .

Here's how to proceed.

$$\begin{array}{c}
 \begin{bmatrix} 1 & 2 & 4 & 3 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix} \\
 \text{apply } r \\
 \downarrow \\
 \begin{bmatrix} 6 & 1 & 2 & 4 & 3 & 5 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix} \\
 \text{apply } (r \cdot m)^4 \\
 \downarrow \\
 \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}
 \end{array}$$

and the puzzle is solved!

Note that if you undo the moves used above you would start with a solved puzzle and end up with all the pieces in their correct positions except that pieces 3 and 4 have been switched. And by rotating first, we could switch any two adjacent pieces on the top that we wish. Similarly we can switch any piece, call it T, from the top with any piece, call it B, from the bottom. First use a rotation and twist to put the two pieces T and B adjacent to each other on the top. Then proceed to switch T and B as described above and finally reverse the twist and rotation used in the first step to move T to B's original location and B to T's.

Modeling the UFO on a TI-89 or TI-92

One can actually implement this matrix model of Rubik's UFO puzzle on a TI-89 or TI-92 graphing calculator. We represent the numbered pieces of the puzzle as being in a column 12-vector (which is a 12 by 1 matrix on the calculator) so that the solved state of the puzzle would look like

$$S = \begin{bmatrix} 1 \\ 2 \\ 3 \\ \vdots \\ 12 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ \vdots \\ 6 \\ 7 \\ 8 \\ \vdots \\ 12 \end{bmatrix} = \begin{bmatrix} S[1,1] \\ S[2,1] \\ \vdots \\ S[6,1] \\ S[7,1] \\ S[8,1] \\ \vdots \\ S[12,1] \end{bmatrix} = \begin{bmatrix} \textit{Top} \\ \textit{Bottom} \end{bmatrix}$$

Then applying r , a 60 degree counter clockwise rotation of the top, can be modeled by

multiplying the vector S on the left by the corresponding permutation matrix which we will also call r . Partitioning the matrices as below, one sees that the bottom is left alone, while the pieces in the top are rotated appropriately.

$$rS = \left[\begin{array}{cccccc|c} 0 & 0 & 0 & 0 & 0 & 1 & \\ 1 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 1 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 1 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 1 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 1 & 0 & \\ \hline & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \end{array} \right] \begin{array}{l} 0 \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ I_6 \end{array} \left[\begin{array}{c} \textit{Top} \\ \hline \textit{Bottom} \end{array} \right] = \begin{array}{c} 6 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ \hline \textit{Bottom} \end{array}$$

The trouble is that the matrices r and S do not fit on the calculator display screen nicely. S is 12 by 1 and r is 12 by 12 and though the calculator has no trouble manipulating matrices of those sizes, it can display only a portion of a large matrix at a time.

However, if we define the matrix $SS = \begin{bmatrix} S[1,1] & S[2,1] & \cdots & S[6,1] \\ S[7,1] & S[8,1] & \cdots & S[12,1] \end{bmatrix}$ on the

calculator, then SS is tied dynamically to the column matrix S . In other words, the result of any action on the entries of S are automatically reflected in the entries of SS .

Thus, once the matrices r , S , and SS are defined in the calculator as described as above, we can rotate the top row of the "puzzle" on the calculator as follows: (A \bullet means calculator input and printed below is the corresponding calculator output.)

- SS (To see the contents of SS .)

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$$

- $r * S \rightarrow S : SS$ (Multiply r times S and store the result in S , then display SS .)

$$\begin{bmatrix} 6 & 1 & 2 & 3 & 4 & 5 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$$

After entering permutation matrices corresponding to the twists, v and x described above then one defines and applies the sequence of moves, m on the calculator with the entries:

- SS (To see the contents of SS .)

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$$

- $v * r * v * r^4 * x * r * v \rightarrow m$ (Stores the given product in the matrix m .)
- $m * S \rightarrow S:SS$ (Multiply m times S and store the result in S , then display SS .)

$$\begin{bmatrix} 1 & 2 & 4 & 3 & 5 & 6 \\ 7 & 8 & 10 & 9 & 11 & 12 \end{bmatrix}$$

Algebraic connections

The permutations of the puzzle pieces which can be accomplished using physical rotations and twists of the UFO form a finite group, call it $ufoGroup$. As described above, we can use manipulations of the puzzle to switch adjacent pieces on the top or on the bottom and to interchange any piece on the top with any piece on the bottom. Thus it is clear that $ufoGroup$ is the symmetric group S_{12} , the group of all possible permutations of 12 pieces.

Here are some suggested activities and questions one could ask an abstract algebra class.

If one starts with a solved puzzle and then rotates the top 6 times, using r , the puzzle returns to the solved position. Applying the twist, v , twice will return a solved puzzle to the solved state. And of course applying the sequence of motions, m , twice will also return a solved puzzle to the solved position since m is really just the product of the transpositions $(3,4)(9,10)$. The minimum number of repetitions of the sequence of motions needed to return a solved puzzle to the solved position is known as the order of the permutation that represents the sequence of motions. So r has order 6 and each of m and v have order 2.

For example, if we begin with a solved puzzle and then rotate, twist and rotate as below the puzzle will be mixed up as shown.

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \end{bmatrix}$$

↓

apply $r \cdot x \cdot r$

↓

$$\begin{bmatrix} 5 & 6 & 10 & 9 & 8 & 4 \\ 7 & 3 & 2 & 1 & 11 & 12 \end{bmatrix}$$

Using an actual puzzle or the calculator model, one can demonstrate that repeating this rotate, twist and rotate action a total of twenty times would mix up a solved puzzle and return it to its solved position.

How could one have predicted that it would take 20 repetitions? That is, how could we have known that $r \cdot x \cdot r$ has order 20?

The sequence of moves $r \cdot x \cdot r$ may be represented as a product of disjoint cycles as $(1, 10, 3, 8, 5)(2, 9, 4, 6)$ and since this is a product of a 5 cycle and a 4 cycle, its order is the least common multiple of 4 and 5, or 20.

Could we find a sequence of moves which would return a solved puzzle to the solved state after 10 repetitions? What is the largest possible order of a permutation that represents a sequence of twists and turns of the puzzle?

Would it be possible to find a more efficient sequence of moves to apply to a solved puzzle which would switch pieces 3 and 4, switch pieces 9 and 10 and leave the others fixed as m does? In other words, could we get the move $m = (3,4)(9,10)$ in fewer twists and turns? One can show the answer to this last question is no by using the algebra software GAP. [3]

Below we explore the group of permutations of the puzzle pieces using GAP and brute force. We begin with the identity and then apply the basic motions r , x , v and t and their inverses to the identity to get the set of permutations, oneMoveAway, whose members are simply one move away from the identity. Applying the basic motions and their inverses to the elements of oneMoveAway and then casting out the permutations that are one move or fewer away from the identity gives the set of elements, twoMovesAway. Continuing in this fashion, we find that there are 64,582 permutations that are eight moves away from the identity and that one of these is $m = v \cdot r \cdot v \cdot r^4 \cdot x \cdot r \cdot v = v \cdot r \cdot v \cdot r^{-1} \cdot r^{-1} \cdot x \cdot r \cdot v = (3,4)(9,10)$. Thus this basic move, m , which we found helpful in solving the UFO puzzle can be made in eight twists and turns of the puzzle and no fewer.

Here is the GAP session: (Comments follow the # symbols.)

```
gap> r := (1,2,3,4,5,6);           # r is a sixty degree counter clockwise rotation.
gap> v := (1,9)(2,8)(3,7);        # v is a vertical twist.
gap> x := (2,10)(3,9)(4,8);       # x is a twist about the sixty degree axis.
gap> t := (3,11)(4,10)(5,9);      # t is a twist about the one hundred twenty
                                  # degree axis.
gap> ri := r^-1;                  # ri is the inverse of r or a sixty degree
                                  # counter clockwise rotation.
                                  # Note that v, x and t are their own inverses.
gap> zeroMovesAway := [()];       # () is the identity permutation.
gap> Size( zeroMovesAway );       # Size determines the size of the set.
1                                  # Only the identity is zero moves from the identity.
gap> soFar := zeroMovesAway;      # We initialize the sets soFar and
gap> prev := zeroMovesAway;       # prev to contain only the identity.
gap> soFar := Union( r*soFar, ri*soFar, v*soFar, x*soFar, t*soFar, soFar);
                                  # Union is set union and Difference is set difference
gap> oneMoveAway := Difference( soFar, prev); # so the set difference soFar \ prev contains
```

```

gap> Size( oneMoveAway );           # the permutations one move from the identity.
5                                   # Size shows there are 5 permutations one move
                                   # away from the identity. They are r, ri, v, x and t.
gap> prev := soFar;                 # Repeating this process we see that
gap> soFar := Union( r*soFar, ri*soFar, v*soFar, x*soFar, t*soFar, soFar);
gap> twoMovesAway := Difference( soFar, prev);
gap> Size( twoMovesAway );
20                                  # there are 20 permutations two moves away from
                                   # the identity.

gap> prev := soFar;
gap> soFar := Union( r*soFar, ri*soFar, v*soFar, x*soFar, t*soFar, soFar);
gap> threeMovesAway := Difference( soFar, prev );
gap> Size( threeMovesAway );
79                                  # There are 79 permutations three moves away from
                                   # the identity.

( ... and so on for fourMovesAway, etc. )
gap> eightMovesAway := Difference ( soFar, prev );
gap> Size( eightMovesAway );
64582                               # There are 64,582 permutations eight moves away
                                   # from the identity.

gap> (3,4)(9,10) in eightMovesAway;
true                                 # Shows that m is eight moves from the identity.
gap> ufoGroup := Group ( r, v, x, t); # ufoGroup is the group generated by the basic
                                   # moves.
Group([ (1,2,3,4,5,6), (1,9)(2,8)(3,7), (2,10)(3,9)(4,8), (3,11)(4,10)(5,9) ])
gap> Size(ufoGroup);
39919800                             # ufoGroup has an order of 39,919,800
                                   # or 11 factorial elements.

```

At the end of the session, GAP agrees that every permutation of the 12 UFO puzzle pieces is possible, just as we had argued above. The group `ufoGroup` defined in the GAP session is the group generated by the basic rotation r , its inverse ri and the three twists v , x , and t . Each of these motions would leave piece number 12 fixed so there would be 11 factorial possible rearrangements of the remaining 11 pieces. Since GAP shows that the order of `ufoGroup` is in fact 11 factorial, each of the possible rearrangements of those 11 pieces can be accomplished by an appropriate sequence of rotations and twists using r , x , v and t .

Clearly when we mix up the puzzle, we permute its pieces. But it's not really that simple. If we simply flip the puzzle over we haven't "mixed up" the puzzle any further yet we would have a different permutation of the pieces. Flipping the pieces on top to the bottom (and being careful not to rotate too) could be thought of as merely interchanging the two rows in the matrix representation of the puzzle. So each mixed up configuration of the puzzle is in some sense equivalent to exactly one of the elements of `ufoGroup`. A given configuration corresponds to the element of `ufoGroup` obtained by merely flying the UFO (by turning or flipping the whole spaceship) so that piece number 12 is in its correct position. This mixed up puzzle could then be

solved by applying the inverse of the corresponding element of `ufoGroup` to the UFO puzzle. This idea, along with the subsets `oneMoveAway`, `twoMovesAway`, `threeMovesAway`, ... defined in the GAP session will provide the most efficient algorithm, known to some as God's algorithm, for solving the UFO puzzle.

It turns out that every element of `ufoGroup` is in exactly one of the sets `oneMoveAway`, `twoMovesAway`, `threeMovesAway`, ... , `sixteenMovesAway`. So every mixed up UFO puzzle can be solved, in some sense, in sixteen moves or less! We can solve an arbitrary mixed up puzzle "most efficiently" by first holding the puzzle so that piece 12 is in its correct position and finding the corresponding element of `ufoGroup` which we will call p . Exactly one of the sets `oneMoveAway`, `twoMovesAway`, `threeMovesAway`, ... , `sixteenMovesAway` will contain p . Let's assume, for example, that p is an element of `fiveMovesAway`. Then at least one of the elements r^*p , ri^*p , v^*p , x^*p , and t^*p must be in `fourMovesAway`. Picking one and applying r , ri , v , x , and t to it will yield an element of `threeMovesAway`. Since `zeroMovesAway` contains only the identity, continuing in this fashion we will have constructed a sequence of basic moves which when applied to p yields the identity. So that sequence, of 5 moves in our example, would solve the puzzle.

When using this algorithm we are sometimes faced with a choice. Making a different choice will lead to a different minimal sequence of moves to solve the puzzle. As an example, one can use GAP to show that the mixed up puzzle

$$\begin{aligned}
 p = (1,5,2,4,8,6,10,7,11,3) &= \begin{bmatrix} 3 & 5 & 11 & 2 & 1 & 8 \\ 10 & 4 & 9 & 6 & 7 & 12 \end{bmatrix} \\
 &= t \cdot v \cdot r \cdot x \cdot t \cdot x \cdot r^{-1} \cdot x \cdot t \\
 &= v \cdot x \cdot t \cdot r \cdot v \cdot t \cdot v \cdot r^{-1} \cdot v
 \end{aligned}$$

is an element of `nineMovesAway` with two very different looking representations. Then to solve the corresponding mixed up puzzle using the first representation, we would apply the basic motions t , v , ri , x , ..., t in turn to get the identity (remember v , x , and t are their own inverses). Using the second representation we would have a very different looking solution but the same number of moves.

How one counts moves is a matter of taste. We have chosen the "sixth turn metric" where we think of r^*r as two moves rather than as a single move of rotating the top of the puzzle 120 degrees. If one counts each rotation of the top, through any multiple of 60 degrees, as a single move (known as the "face turn metric") every mixed up puzzle may be solved in at most 14 moves. See [4] for more details. There (and as Sequence A079820 in the On-Line Encyclopedia of Integer Sequences [5]) you will find the sizes of the sets `zeroMovesAway`, `oneMoveAway`, ..., `sixteenMovesAway` which we used GAP to find.

Though the order of `ufoGroup` = 39,916,800 is huge, it is small compared to the corresponding group for Rubik's cube which has $8! \cdot 2! \cdot 3^7 \cdot 2^{10} = 43,252,003,274,489,856,000$ or $4.3 \cdot 10^{19}$ possible positions. This group is so large that a similar brute force computer implementation of God's algorithm is impossible. At present, we know there are solution methods for the cube that take at most 42 ninety degree face turns to solve the cube from any given starting position, and that there is a position of the cube that takes at least 26 ninety degree face turns to solve.

References

- [1] Adventures in Group Theory: Rubik's Cube, Merlin's Machine, and Other Mathematical Toys. David Joyner. Johns Hopkins University Press, 2002
- [2] Herbert Kociemba's Cube Explorer - <http://www.kociemba.homepage.t-online.de/cube.htm>
- [3] GAP System for Computational Discrete Algebra - <http://www.gap-system.org>
- [4] Jaap Scherphuis' Rubik's UFO page - <http://www.geocities.com/jaapsch/puzzles/rubufo.htm>
- [5] On-Line Encyclopedia of Integer Sequences - <http://www.research.att.com/~njas/sequences/Seis.html>