Modified Kociemba Two Phase Method

Tsung-Wu Lin 20230615

Introduction

1. 8 corners, and each has 3 orientations. The number of corner orientation states is 3^7=2187. The number of corner permutation states is 8!=40320.

2. 12 edges, and each has 2 orientations. The number of edge orientation states is 2^11=2048. The number of edge permutation states is 12!= 479,001600.

3. The number of the possible states is 2187*2048*40320*479001600/2=43,252003,274489,856000.

The reason for the number of corner/edge orientation states is 3^7/2^11, instead of 3^8/2^12: Each rotation always changes 2 corners/edges with opposite rotation directions, therefore, it is impossible to have a single corner/edge with incorrect orientation. So, if 7-corners/11-edges are orientated then 8-corners/12-edges must be orientated. 1/2 is for the number of permutations that must be even.

Kociemba Two Phase Method

1. Use G1 as an intermediate state.
2. From G1 state, it can use the moves in G1={F2,R2,B2,L2,U,D} to complete the cube. And it is called Phase 2. Each move in Phase 2 will keep the cube remain in G1 state. And to complete the cube only requires at most 18 steps.
3. From any random state, it can use all 18 moves in G0={F,R,B,L,U,D} to bring the cube into the G1 state. And it is called Phase 1. To get the G1 state, Phase 1 only requires at most 12 steps.
4. G1 state is defined as: (1) The color of up/down faces must be U/D color. (2) In Mid-layer, the color of the front/back faces must be F/B color. (3) In Mid-layer, the color of the left/right faces must be L/R color. (Whenever (1) and (2) are satisfied, (3) will be satisfied)
5. In G1 state: The orientations of 8-corners/12-edges are correct. 4-edges in the mid-layer remain in that layer. The mid-layer is also called the second layer or the UD slice.
6. The number of states in G1 = 8! * 8! * 4!/2 = 40320 * 40320 * 12 = 19508,428800.

Modified Kociemba Two Phase Method:

1. Use G1' as an intermediate state.
2. The number of states in G1' = 8! * 8!/16 * 4!/2 = 40320 * 2520 * 12 = 1219,276800 = 1/16 of that in G1. The difference between G1' and G1 is: Apply any U/D turn on the Up/Down layer remain the cube in the same state in G1'. Therefore, the number of states in G1' is only 1/16 of that in G1. The states can be represented by a 32bits standard integer number.
3. From G1' state, it can use the moves in G1={F2,R2,B2,L2,U,D} to complete the cube. And it is still called Phase 2. Each move in Phase 2 will keep the cube remain in G1' state. And to complete

the cube only requires at most 17 steps. But it may need one U-turn and/or one D-turn to match the position from the solution table.

4. From any random state, it can use all 18 moves in G0={F,R,B,L,U,D} to bring the cube into the G1' state. To get the G1' state, Phase 1 only requires at most 12 steps.
5. In Phase 1 to bring the cube to G1' state, the orientations (3^7/2^11) of 8-corners/12-edges and the permutation (12!/8!/4!) of 4-edges in the mid-layer are computed to identify one of the cube states of 3^7 * 2^11 * (12!/8!/4!) =2187*2048*495=2217,093120 for the solution table in p0[]. The cube states of 2217,093120 can be represented by a 32bits standard integer number.

Considerations in Computer program:

The 6 faces F,R,U,B,L,D are named as 1,2,3,4,5,6 and the corresponding color are also named as 1,2,3,4,5,6. Use a vector S[54] to store the color of 54 cube faces. The S[1:54] for a solved cube is

3,1,2, 3,2,4, 3,4,5, 3,5,1, 6,2,1, 6,4,2, 6,5,4, 6,1,5,
3,1, 3,2, 3,4, 3,5, 6,1, 6,2, 6,4, 6,5, 1,2, 4,2, 4,5, 1,5, 1, 2, 3, 4, 5, 6.

The first 24 colors are for the 8 corners. 3 faces count-clock-wise with the up/down face going first.

The next 24 colors are for the 12 edges. 2 faces in the up/down layer with the up/down face going first. 2 faces in the mid-layer with the front/back face going first.

The last 6 colors are for the center cube face. Since they don't change for any moves, they are not used.

Phase one

Each move rotates 20 cube faces and rotates their face colors. Use a rotation about face 1(F) as an example: The indexes of S[] for the faces to be rotated are {1, 2, 3,25,26, 11,12,10,48,47, 22,23,24,33,34, 14,15,13,42,41}. For a 90-degree CCW rotation the color codes in 1, 2, 3,25,26 positions are moved to 11,12,10,48,47 positions; 11,12,10,48,47 positions are moved to 22,23,24,33,34 positions; 22,23,24,33,34 positions are moved to 14,15,13,42,41 positions; 14,15,13,42,41 positions are moved to 1, 2, 3,25,26 positions. Please see function ROTATEA() for details.

Based on the 3 colors of the i-th (1:8) corner, the corner number j=1:8 and the rotation k[i]=0:2 can be obtained, and let x[i]=10*j+k[i]. Based on the 2 colors of the i-th (9:20) edge, the edge number j=9:20 and the rotation k[i]=0:1 can also be obtained, and let x[i]=10*j+k[i].

Based on the rotations k[1:7] of the first 7 corners to get a corner rotation state icc=0:2186 as Icc = (((((k[1]*3+k[2])*3+k[3])*3+k[4])*3+k[5])*3+k[6])*3+k[7].

Based on the rotations k[9:19] of the first 11 edges to get an edge rotation state iee=0:2047 as iee = ((…((k[9]*2+k[10])*2+k[11])*2+k[12])*2+k[13])*2+k[14])*2+k[15])*2+k[16])*2+k[17])*2+k[18])*2+k[19].

Based on the permutations of the 4 edges in the UD slice to get a UD slice permutation state ipp=0:494.

And combine them to get a cube state p0=0:2217,093119 as p0 = (icc*2048+iee)*495+ipp.

In phase one, a solution table m0[95039] and the corresponding state table p0[95039] are used. These tables are set up using at most 5 steps of G0={F,R,B,L,U,D}. Then Phase one searches at most 7 steps of G0={F,R,B,L,U,D}. to check the resulting xp0 if it matches the states in the p0[95039] table and get the solution m0[] to bring the cube to a G1' state.

Note that the 5 steps of G0={F,R,B,L,U,D} can reach 277388 states but only 95039 are different.

Phase one begins by searching the first 18 one-step states, then searching the next 108 two-step states, then searching the 1440 three-step states, and so on for the 19224 four-step states, the 256608 five-step states, the 3425328 six-step states, and the last 45722880 seven-step states.

Phase two

In Phase 2, the orientations k[i] of the corners and edges are correct and not needed. This phase is used to permutate the corners and the edges to the correct sequence. And then let x[i] = j in this phase.

The x[1:20] for a solved cube are: 1,2,3,4, 5,6,7,8, 9,10,11,12, 13,14,15,16, 17,18,19,20.

1,2,3,4 are the 4 corners at the up layer, and the colors are 3,1,2; 3,2,4; 3,4,5; 3,5,1.
5,6,7,8 are the 4 corners at the down layer, and the colors are 6,2,1; 6,4,2; 6,5,4; 6,1,5.
9,10,11,12 are the 4 edges at the up layer, and the colors are 3,1; 3,2; 3,4; 3,5.
13,14,15,16 are the 4 edges at the down layer, and the colors are 6,1; 6,2; 6,4; 6,5.
17,18,19,20 are the 4 edges at the UD slice, and the colors are 1,2; 4,2; 4,5; 1,5.

Each move rotates 4 corners and 4 edges. Use a rotation about face 1(F) as an example: The indexes of x[] for the corners/edges to be rotated are {1,5, 4,20, 12,13, 9,17}. For a 90-degree CCW rotation, the corner/edge in 1,5 positions are moved to 4,20 positions; 4,20 positions are moved to 12,13 positions; 12,13 positions are moved to 9,17 positions; 9,17 positions are moved to 1, 5 positions. Please see function ROTATEB() for details.

Based on the permutations (8!) of the 8 corners to get a corner permutation state pcc=0:40319.

Based on the permutations (8!) of the 8 edges in the up/down layers, get an edge permutation state pee=0:40319. This state, pee, is reduced to ipp=0:2519, for only relative positions are considered.

Based on the permutations (4!) of the 4 edges in the UD slice to get a UD slice permutation state kee=0:23.

And combine them to get a cube state p1=0: 2438553599 as p1 = (ipp*40320+pcc)*24+kee.

In phase two, a solution table m1[295325] and the corresponding state table p1[295325] are used. These tables are set up using at most 7 steps of G1={F2,R2,B2,L2,U,D}. Then Phase two searches at most 10 steps of G1={F2,R2,B2,L2,U,D} to check the resulting xp1 if it matches the states in the p1[295325] table and get the solution m1[] to complete the cube.

Note that the 7 steps of G1={F2,R2,B2,L2,U,D} can reach 544907 states but only 295325 are different.

Phase two begins by searching the first 10 one-step states, then searching the next 34 two-step states, then searching the 216 three-step states, and so on for the 1492 four-step states, the 10088 five-step states, the 68456 six-step states, the 464616 seven-step states, the 3151712 eight-step states, the 21386292 nine-step states, and the last 145099168 ten-step states.

The cube state G1’ is computed based on a specific position at the up/down layer for the edges with the maximum edge number. Therefore, it may need to add a U-turn and/or a D-turn to match the position in the solution table.

Appendix:

G0={F,R,B,L,U,D}:18

G1={F2,R2,B2,L2,U,D}:10

G2={F2,R2,B2,L2}:4

G3=G0-G1={F,F’,R,R’,B,B’,L,L’}:8

G4=G1-G2={U,D}:6

```
======Phase ONE=====   =========Phase TWO========
--Search--  ---Table p0---   ----Search----            ---Table p1---
G0::::::G0  G0::::::G0:G3  G1::::::G1:G2  G4:G4  G2:G1::::::G1
                  G3+G1 = G0                  G1       G1 = G4+G2
```

Phase one: Search 7 steps of G0 for xp0 in Table p0[95039]. And get the move in m0[].

Phase two: Search 10 steps of G1 with the last step G2 for xp1 in Table p1[295325].
And get the move in m1[]. Adjust by two G4 steps to match the U & D positions.

Table 1. The number of elements obtained by the steps in (a) Phase One, and (b) Phase Two

S(steps)	E(elements) in Phase One
0	1
1	5
2	55
3	647
4	7803
5	95039
6	1138856 (0.05%)
7	13209134 (0.6%)
8	138155502 (6.2%)
9	959761462 (43%)
10	2158890200 (97%)
11	2217092644 (99.999978%)
12	2217093120 (100%)
	= (3^7)*(2^11)*(12!/8!/4!)
	= 2187*2048*495 = 2217093120
	= Total elements in G0

S(steps)	E(elements) in Phase Two
0	1
1	5
2	39
3	253
4	1570
5	9357
6	54830
7	295325 (0.02%)
8	1519971 (0.12%)
9	7573048 (0.62%)
10	34992638 (3.0%)
11	141552026 (12%)
12	433575844 (35%)
13	861474530 (70%)
14	1169329308 (96%)
15	1218528172 (99.9386%)
16	1219276224 (99.999953%)
17	1219276800 (100%) = Total elements in G1'
	= (8!8!/16*4!)/2 = 2438553600/2 = 1219276800
	: (1/2 for even permutation only)

G3=G0-G1={F, F’,R, R’,B, B’,L, L’}: For the first step to setup the table p0[].

G2={F2,R2,B2,L2}: For the Last step to setup the table p1[] and in the search to match the table p1[].

G1 =(F2,R2,B2,L2,U,D): Total elements in G1 = 1219276800*16 (too big)

The Two-Phase Algorithm: G0 * G1 : 2217093120 * 1219276800*16

Modified Two-Phase Algorithm: G0 * G1' : 2217093120 * 1219276800