修訂 Kociemba Two Phase Method

林聰悟 20230615

簡介

1. 8 個角塊，各有 3 個方向。方向組合共有 3^7=2187 個。排列組合共有 8!=40320 個。

2. 12 個邊塊，各有 2 個方向。方向組合共有 2^11=2048 個。排列組合共有 12!=479,001600 個。

3. 所有可能的狀態為 2187*2048*40320*479001600/2= 43,252003,274489,856000個。

8 個角塊之方向組合共有 3^7=2187 個而不是 3^8 個，12 個邊塊之方向組合共有 2^11=2048 個而不是 2^12 個，是因為每次轉動會同時改變二個角塊之方向且一個正轉另一個反轉，以及會同時改變二個邊塊之方向，故沒有單一角塊或單一邊塊方向不對之情形。所以若 7 個角塊或 11 個邊塊之方向都對就保證 8 個角塊或 12 個邊塊之方向都對。1/2 是因為排列組合須得自偶數次交換。

Kociemba Two Phase Method

1. 以 G1 狀態為中繼站。
2. 由 G1 狀態可僅用 G1={F2,R2,B2,L2,U,D}而完成方塊，此階段稱為 Phase two。每個轉動都會留在 G1 狀態。Phase two 最多轉 18 次一定可以完成方塊。
3. 由任意之混亂狀態用所有轉法，即 G0={F,R,B,L,U,D}或 6 面各 3 種角度共 18 種轉法，最多轉 12 次一定可以轉到 G1 狀態。此階段稱為 Phase one。
4. G1 狀態為：(1)上層上面與下層下面的顏色必須為 U 或 D 色。(2)中層前後面的顏色必須為 F 或 B 色。(3) 中層左右面的顏色必須為 R 或 L 色。(當(1)與(2)滿足時， (3)會自動滿足)
5. G1 狀態下：8 個角塊的方向都正確，12 個邊塊的方向也都正確。4 個中層的邊塊都會留在中層內。中層又稱為 UD 夾層或第二層。
6. G1 狀態共有 8! * 8! * 4!/2 = 40320 * 40320 * 12 = 19508,428800 個。

Modified Kociemba Two Phase Method:

1. 以 G1’ 狀態為中繼站。
2. Phase two 之 G1’ 狀態共有 8! * 8!/16 * 4!/2 = 40320 * 2520 * 12 = 1219,276800 個為 G1 者的 16 分之 1。
   G1’ 與 G1 狀態的差異為：上層的排列用{U}轉動後均視為同一種 G1’ 狀態。下層的排列用{D}轉動後亦均視為同一種 G1’ 狀態。因此 G1’ 狀態總數只有 G1 者的 16 分之 1。大大減少狀態數。狀態標號由 19508,428800 降至 1219,276800，而可用 32bits 的標準整數編號。
3. Phase two 由 G1’ 狀態僅用 G1={F2,R2,B2,L2,U,D}而完成方塊。每個轉動都會留在 G1’ 狀態。最多轉 17 次也一定可以完成方塊。但因 G1’ 狀態總數只有 G1 者的 16 分之 1。上面與下面可能各須做一次{U}或{D}轉動(即 G4={U,D})使其對位。
4. Phase one 則是由任意給定之混亂狀態用所有轉法，即 G0={F,R,B,L,U,D} (6 面各 3 種角度共 18 種轉法)，最多轉 12 次一定可以轉到 G1’狀態。
5. G1={F2,R2,B2,L2,U,D}的轉動情形有 10 種(F2,R2,B2,L2,U,U’,U2,D,D’,D2)，排除不與前次轉動同一面而少 1 或 3 種。平均約為 8 種((4*9+6*7)/10=7.8)。

6. Phase one 轉往 G1’ 狀態的重點在：8 個角塊的方向要正確(3^7)；排列順序不拘。12 個邊塊的方向也要正確(2^11)；4 個中層的邊塊要落在中層(12!/8!/4!)。因此中間的狀態編碼數為：3^7 * 2^11 * (12!/8!/4!) =2187*2048*495=2217,093120。此數亦可僅用 32bits 的標準整數編號。

電腦程式之處理方式：

F,R,U,B,L,D 六面之編號用 1,2,3,4,5,6，其對應顏色之編碼亦為 1,2,3,4,5,6。

用一維陣列 S[54]儲存方塊之 54 個表面之顏色碼。正確的方塊顏色碼為:

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

前 24 個為 8 個角塊各 3 表面之顏色碼。3 表面按反時鐘方向之順序排列，上面或下面優先。

其後 24 個為 12 個邊塊各 2 表面之顏色碼。上下層以上面或下面優先。中層以前面或後面優先。

最後 6 個為 6 個中心塊各 1 表面之顏色碼 (這 6 個顏色碼因不受轉動影響而改變，可以不用)。

Phase One

每次轉動須輪轉 20 個表面之顏色碼。以方塊 1 面(F)之轉動為例，須輪轉之 20 個表面在 S 陣列之位置為：{1, 2, 3,25,26, 11,12,10,48,47, 22,23,24,33,34, 14,15,13,42,41}。將方塊 1 面反轉 90 度之動作為：1, 2, 3,25,26 位置之顏色碼移到 11,12,10,48,47 位置；11,12,10,48,47 位置之顏色碼移到 22,23,24,33,34 位置；22,23,24,33,34 位置之顏色碼移到 14,15,13,42,41 位置；14,15,13,42,41 位置之顏色碼移到 1, 2, 3,25,26 位置。詳見函數 ROTATEA。

根據 8 個角塊位置(編號 i=1:8)之顏色碼可以確定其來自那一個角塊(j=1:8)，以及其轉向(k[i]=0,1,2)。以角塊碼*10+轉向碼合併為單一碼並儲存在陣列 x。即令 x[i]=j*10+k[i]。

根據 12 個邊塊位置(i=9:20)之顏色碼可以確定其來自那一個邊塊(j=9:20)，以及其轉向(k[i]=0,1)。以邊塊碼*10+轉向碼合併為單一碼並儲存在陣列 x。即令 x[i]=j*10+k[i]。

根據各角塊轉向碼可組合為單一角塊轉向碼 icc。根據各邊塊轉向碼可組合為單一邊塊轉向碼 iee。根據 UD 夾層各邊塊碼可組合為單一夾層邊塊排列碼 ipp。

icc=(((((k[1]*3+k[2])*3+k[3])*3+k[4])*3+k[5])*3+k[6])*3+k[7] 介於 0:2186。

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] 介於 0:2047。

ipp=0:494

最後合併為 Phase one 狀態碼 p0=(icc*2048+iee)*495+ipp 。p0 介於 0: 2217,093119 可僅用 32bits 的標準整數表示。

Phase one 之解法採用最多轉 5 步之方塊狀態表 p0[95039]，共有 95039 個答案及對應之轉法 m0[95039]。搜尋答案最多試轉 7 次。轉 5 步內之方塊狀態共有 277388 個但僅 95039 個不相同。

亦可用最多轉 6 步，7 步或 8 步之答案表 p0[1138856]，p0[13209134]或 p0[138155502]，可以減少搜尋答案之試轉次數而加快求解速度，但需要較多存資料之記憶體。

搜尋答案採用廣度優先策略：先檢驗所有走一步共有 18 種轉法之方塊狀態是否在答案表內，若有則由其對應之轉法即可轉到 G1'方塊狀態，而完成 Phase one。否則再檢驗所有走二步共有 108 種轉法之方塊狀態是否在答案表內。否則再檢驗所有走 3 步共有 1440 種轉法之方塊狀態是否在答案表內。依此類推走 4 步共有 19224 種轉法，走 5 步共有 256608 種轉法，走 6 步共有 3425328 種轉法，走 7 步共有 45722880 種轉法。

Phase Two

進入 Phase two 後角塊與邊塊之方向都已正確，Phase two 基本上為調整角塊之排列順序，與調整邊塊之排列順序。因此方塊之狀態可改用一維陣列 x[20]儲存 20 個方塊位置是來自那一個方塊。正確的方塊碼為: 1,2,3,4, 5,6,7,8, 9,10,11,12, 13,14,15,16, 17,18,19,20。

1,2,3,4 為上層 4 個角塊其顏色為 3,1,2; 3,2,4; 3,4,5; 3,5,1。
5,6,7,8 為下層 4 個角塊其顏色為 6,2,1; 6,4,2; 6,5,4; 6,1,5。
9,10,11,12 為上層 4 個邊塊其顏色為 3,1; 3,2; 3,4; 3,5。
13,14,15,16 為下層 4 個邊塊其顏色為 6,1; 6,2; 6,4; 6,5。
17,18,19,20 為 UD 夾層 4 個邊塊其顏色為 1,2; 4,2; 4,5; 1,5。

每次轉動須輪轉 4 個角塊與 4 個邊塊。以方塊 1 面(F)之轉動為例，須輪轉之 8 個角塊與邊塊在 x 陣列之位置為：{1,5, 4,20, 12,13, 9,17}。將方塊 1 面反轉 90 度之動作為：1,5 位置之方塊碼移到 4,20 位置；4,20 位置之方塊碼移到 12,13 位置；12,13 位置之方塊碼移到 9,17 位置；9,17 位置之方塊碼移到 1,5 位置。詳見函數 ROTATEB。

根據 8 角塊位置之方塊碼可組合為單一角塊排列碼 pcc=0:40319。根據上層 4 邊塊與下層 4 邊塊位置之方塊碼可組合為單一上下層邊塊排列碼 pee=40319。根據 UD 夾層 4 邊塊位置之方塊碼可組合為單一夾層邊塊排列碼 kee=0:23。單一上下層 8 邊塊排列碼 pee，若僅考慮相對位置而不計較絕對位置，可由 40320 種組合降低為 2520 種組合，排列碼 pee 改為 ipp=0:2519。

最後合併為 Phase two 狀態碼 p1=(ipp*40320 + pcc)*24 + kee 。p1 介於 0: 2438553599，亦可僅用 32bits 的標準整數表示。(註：8!=40320，4!=24，40320/16=2520，2^32=4294967296)

Phase two 之解法類似 Phase one 採用最多轉 7 步之方塊狀態表 p1[295325]，共有 295325 個答案及對應之轉法 m1[295325]。搜尋答案最多試轉 10 次。轉 7 步內之方塊狀態共有 544907 個但僅 295325 個不相同。

亦可用最多轉 8 步，9 步或 10 步之答案表 p1[1519971]，p1[7573048] 或 p1[34992638]，可以減少搜尋答案之試轉次數而加快求解速度，但需要較多存資料之記憶體。

搜尋答案亦採用廣度優先策略：先檢驗所有走一步共有 10 種轉法之方塊狀態是否在答案表內，若有則由其對應之轉法即可轉到正確方塊狀態，而完成 Phase two。否則再檢驗所有走二步共有 34 種轉法之方塊狀態是否在答案表內。否則再檢驗所有走 3 步共有 216 種轉法之方塊狀態是否在答案表內。依此類推走 4 步共有 1492 種轉法，走 5 步共有 10088 種轉法，走 6 步共有 68456 種轉法，走 7 步共有 464616 種轉法，走 8 步共有 3151712 種轉法，走 9 步共有 21386296 種轉法，走 10 步共有 145099168 種轉法。

為了比對答案表之狀態碼，檢驗之方塊與答案表之方塊均各須轉動上層 ju*90 度與 iu*90 度及或轉動下層 jd*90 度與 id*90 度到一特別位置(定為上層編號最大之邊塊轉置於第 4 邊塊位置，下層編號最大之邊塊轉置於第 8 邊塊位置。)

因此 Phase two 答案表之轉法須於上層加轉 ju*90 度；於下層加轉 jd*90 度之後才會是答案表對應之方塊狀態。而搜尋時檢驗之方塊狀態須於上層加轉 iu*90 度；於下層加轉 id*90 度之後才會是答案表對應之方塊狀態。總而言之，檢驗之方塊狀態須於上層加轉(iu-ju)*90 度；於下層加轉(id-jd)*90 度之後才能接續答案表對應的轉法。

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