東邦大学理学部情報科学学科

2015年度

卒業研究論文

群論とルービックキューブ

提出日　2016年1月29日(金)

指導教員　白柳　潔

提出者　　水野貴裕

概要

置換パズルである 3×3×3 のルービックキューブは、群論などを用いて数学的に解析することができる。本研究では、計算機代数システムの SAGE を利用したコンピューターによる解法と Layer By Layer 法(略称 LBL 法)による解法に対して、両者の手数を比較した。

SAGE を利用した解法は、福岡大学の藤本光史教授の研究や David Joyner 著の『群論の味わい』を参考資料とした。ルービックキューブの状態を置換群の元として入力し、キューブの配置を変えてから SAGE を使って初期状態に揃えさせている。

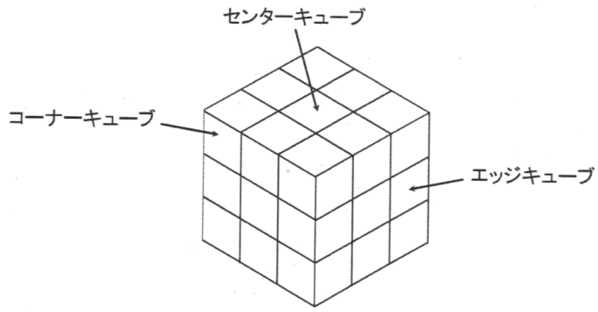
LBL 法はルービックキューブの大会でよく使われており、最速の解法として知られている。ルービックキューブの構造を 3 層と見て、下層から順に複数のキューブを揃えていく方法である。

目次

第１章　序論

1.1　はじめに

ルービックキューブは 1974 年にハンガリーの建築学者エルノー・ルービックが考案した置換パズルであり、8 個のコーナーキューブと、12 個のエッジキューブと、6 個のセンターキューブの計 26 個の小方体で構成された魔方体である。そのすべての小方体の面、54 面の置換群、または動かない 6 面を除いて 48 面の置換群ととらえることができる。



また、センターキューブは１面体、エッジキューブは２面体、コーナーキューブは３面体と言われることもある。

「ばらばらになった状態から 6 面を揃える」ことが基本的な遊びである。

日本では 1980 年にツクダオリジナル(現メガハウス)より発売された。3×3×3 のルービックキューブ以外にも 2×2×2 のルービックキューブ、4×4×4 のルービックキューブリベンジ、5×5×5 のプロフェッサーキューブも次いで発売された。その後もルービックキューブの種類は増えていき、穴の開いたルービックボイドキューブ、回すたびに形の変わるルービックミラーブロックスなどが発売された。

ルービックキューブは多くの人が研究しているが非常に難しい内容であるため計算機を用いて計算されている。本研究では SAGE を利用しコンピューターを用いて求める解法と、人の手による解法の手数を比較し、考察することを目的とする。

SAGE とは自由に利用することのできるオープンソースソフトウェアとして開発された計算機台数システムである。SAGE はダウンロードしてインストールする版だけでなく、インストールしないでオンラインで実行することもできる。

1.2　ルービックキューブについて

ルービックキューブを数学的に表すと一人ゲームの置換パズルである。

一人ゲームとは

- それぞれの局面では、有限個の許される手がある。
- 有限の手の並び(手順)で解に到達できる。
- 偶然に左右されたり、無作為に選ばれる手はない。
- それぞれの手を行うことの効果はすべてわかっている。

・それぞれの手が許されるかどうかは、現在のゲームの状態にだけ依存していて、
それまでの手がどうであったかに依存しない。

以上の条件にあてはまるものである。

置換パズルとは、小片の有限集合 T を使う次の四つの条件を満たす一人ゲームである。

(1)パズルのそれぞれの手は、Z_nの元の置換に対応する。ただし、$n > 1$はパズルの構成によって決まる整数とする。

(2)Z_nの一つの置換に対応する手が複数ある場合、それぞれの手を行った結果の状態は区別できない。

(3)それぞれの手Mは「可逆」つまりMに対してMを行った結果の状態からMを行う前の状態に戻す手が存在する。

(4)Tの置換f_1に対応する手をM_1とし、T の置換f_2に対応する手をM_2とすると$M_1 * M_2$(M_1に引き続いてM_2を行う)は次のどちらかとなる。

・ゲームの規則では許されない手

・置換$f_1 * f_2$に対応する手

次に 3×3×3 のルービックキューブが置換パズルであることを確認するのに必要な表記を紹介する。ルービックキューブは 6 個の面があり、それぞれの面は 3×3＝9 個の小面から構成されるので、全部で 54 面の小面がある。単位操作によって各面の中央の小面は移動しないので 54－6＝48 個の小面にだけ 1，2,……，48 の番号を付ける

			1	2	3						
			4	U	5						
			6	7	8						
9	10	11	17	18	19	25	26	27	33	34	35
12	L	13	20	F	21	28	R	29	36	B	37
14	15	16	22	23	24	30	31	32	38	39	40
			41	42	43						
			44	D	45						
			46	47	48						

互いに素な巡換記法で表したルービックキューブの 6 個の面に対応する標準的な生成元は次の通り。

F = (17,19,24,22)(18,21,23,20)(6,25,43,16)(7,28,42,13)(8,30,41,11)
B = (33,35,40,38)(34,37,39,36)(3, 9,46,32)(2,12,47,29)(1,14,48,24)
L = (9,11,16,14)(10,13,15,12)(1,17,41,40)(4,20,44,37)(6,22,46,35)
R = (25,27,32,30)(26,29,31,28)(3,38,43,19)(5,36,45,21)(8,33,48,24)
U = (1, 3, 8, 6)(2, 5, 7, 4)(9,33,25,17)(10,34,26,18)(11,35,27,19)
D = (41,43,48,46)(42,45,47,44)(14,22,30,38)(15,23,31,39)(16,24,32,40)

小面の表記法はシングマスター記法を使う。立方体の中央に接する正面はxyという形式で表記する。但し、xはその小面が含まれる面を表し、yはその小面に隣接する面を表す。それぞれの面にある小面は右上隅から初めて時計回りに次の通りになる。

前面	fru	fr	frd	fd	fld	fl	flu	fu
後面	blu	bl	bld	bd	brd	br	bru	bu
右面	rbu	rb	rbd	rd	rfd	rf	rfu	ru
左面	lfu	lf	lfd	ld	lbd	lb	lbu	lu
上面	urb	ur	urf	uf	ulf	ul	ulb	ub
下面	drf	dr	drb	db	dlb	dl	dlf	df

SAGE を使って正面のシングマスター記法による表記を求めることができる。そのプログラムは以下のとおりである。例として 41 の小面の表記を求める。

```
sage: from sage.groups.perm_gps.cubegroup import *
sage: index2singmaster(41)
        'dlf'
```

第 2 章　群論

ルービックキューブ群について解析するために必要な群論について紹介する。

2.1　群の一般定義

Gを空でない集合とし、二項演算

$$\begin{array}{ccc} G \times G & \to & G \\ (a,b) & \mapsto & a \cdot b = c \end{array}$$

が与えられていて次の 3 条件を満たす時、G を群(group)という。

(G1) 結合法則

$\forall a, b, c \in G$に対し、$a \cdot (b \cdot c) = (a \cdot b) \cdot c$が成り立つ。

(G2) 単位元の存在

$e \in G$が存在して、$\forall a \in G$に対し、$e \cdot a = a \cdot e = a$が成り立つ。

このeをGの単位元と言う。

(G3) 逆元の存在

$\forall a \in G$に対し、aに対応した元$x \in G$が存在し、 $a \cdot x = x \cdot a = e$が成り立つ。

このxをaの逆元と言う。

2.2 置換群

X を有限集合とする。$\mathrm{S} = \{g_1, g_2, \cdots, g_n\}$を X の置換からなる有限集合とする。G を次の形式のすべての積の集合とする。

$$\mathrm{g} = x_1 * x_2 * \cdots * x_m \quad (m > 0)$$

ただし、$x_1, \cdots, x_m$は S の元とする。置換の合成を群とする集合 G を$g_1, \cdots, g_n$を生成元とする置換群(または S から生成された生成群)という。これを次のように表記する。

$$\mathrm{G} = \langle g_1, g_2, \cdots, g_n \rangle \subset S_X$$

X をルービックキューブの 54 個の小面の集合とし、R,L,U,D,F,B∈ S_Xをルービックキューブ群の単位操作とする。置換群

$$\mathrm{G} = \langle \mathrm{R, L, U, D, F, B} \rangle \subset S_X$$

をルービックキューブ群という。後程ルービックキューブ群の構造を決定する。

2.3 位数

G を群とするとき、G の濃度を G の位数といい、$|\mathrm{G}|$と表記する。群 G の元gに対して$g^m = 1$を満たす最小の正整数mが存在するとき、このmをgの位数と言い、$\mathrm{ord}(g)$と表記する。このような整数mが存在しないとき、gを無限位数の元という。

SAGE を使って、ルービックキューブ群およびその元の位数を調べることができる。そのプログラムは以下の通り。

```
sage: cube=PermutationGroup([f,b,l,r,u,d])
sage: F,B,L,R,U,D=cube.gens()
sage: cube.order()
        43252003274489856000
sage:N.factor
        2^27*3^14*5^3*7^2*11
```

これにより$|G| = 2^{27}3^{14}5^{3}7^{2}11$(約$4.3 \times 10^{19}$)となることが分かる。

2.4　コーシーの定理

(a)p を$|G|$を割り切る素数だとすると、位数 p の元$g \in \mathrm{G}$が存在する。

(b)n を$|G|$を割り切らない整数とすると、位数 n の元$g \in \mathrm{G}$は存在しない。

ルービックキューブ群 G は$|G| = 2^{27}3^{14}5^{3}7^{2}11$となるのでコーシーの定理(b)より位数 13 のルービックキューブの手順は存在しない。なぜなら$|G|$は 13 で割り切れないため。

2.5　部分群

G を群とする。G の部分集合 H で、部分集合として G から引き継いだ演算*が群の条件をすべて満たすものを G の部分群という。

G を群とするとき、「H は G の部分群となる」という命題は H⊂G と表記する。

ルービックキューブ群の部分群は数が多すぎるため、何個なるのか誰もわかっていない。

2.6　交換子

gおよびhを群 G の元とするとき、元

$$[g,h] = g * h * g^{-1} * h^{-1}$$

をgとhの交換子という。gとhが可換な時に限り$[g,h] = 1$になる。

G をS_{54}の元とみなしたR, L, U, D, F, Bから生成される置換群とする。

$$[F, R^{-1}] = F * R^{-1} * F^{-1} * R$$

を Y 交換子という。また、

$$[F, R] = F * R * F^{-1} * R^{-1}$$

を Z 交換子という。

これらの操作はルービックキューブを解くときに頻繁に行う操作である。ルービックキューブの多くの手順は Y 交換子や Z 交換子によって生成される。

2.7　共役

gおよびhを群 G の元とするとき、元

$$g^{h} = h^{-1} * g * h$$

をhによるgの共役という。

gとhが可換の時に限り$g^{h} = g$となる。

この操作はルービックキューブを解くときに頻繁に行う操作である。

2.8　剰余類

G を群とし、H を G の部分群とする。群の演算を乗法で表記し、gを G の元とするとき、

G の部分集合 g*H を G における H の左剰余類といい、G の部分集合 H* gを G における H の右剰余類という。G が可換群の場合、H を G の部分群とすると左剰余群と右剰余群は一致する。

2.9　剰余群

H を G の正規部分群とすると G の元aおよびbに対する二項演算

$$a\mathrm{H}\cdot b\mathrm{H}=(ab)\mathrm{H},\ \ (a\mathrm{H})^{-1}=a^{-1}\mathrm{H}$$

に関して剰余類の集合 G/H は群となる。自明な剰余類 H がこの群 G/H の単位元となる。

この群 G/H を G の H による剰余群、または商群という。

2.10　群の直積

H_1およびH_2を二つの群という。群 G が次の条件を満たすとき、H_1とH_2の直積と言い

$$\mathrm{G}=H_1\times H_2$$

と表記する。

(a)$\mathrm{G}=H_1\times H_2$(集合としての直積となる)

(b)群 G の演算は次の成分ごとの演算で与える。

$$(x_1,y_1)\cdot(x_1,y_1)=(x_1\cdot x_2,y_1\cdot y_2)$$

ただし、$x_1,x_2\in H_1$および$y_1,y_2\in H_2$とする。

2.11　群の半直積

H_1およびH_2を群 G の部分群とする。

G が次の条件を満たすとき、G をH_1とH_2の半直積と言い

$$\mathrm{G}=H_1\rtimes H_2$$

と表記する。

(a)$\mathrm{G}\cong H_1\cdot H_2=\{h_1\cdot h_2|h_1\in H_1,h_2\in H_2\}$

(b) H_1およびH_2に共通の元は G の単位元 1 のみ。

(c) H_1は G の正規部分群

この定義を内部半直線ということがある。

第 3 章　ルービックキューブ群の構造

3.1　キューブ理論の第 1 基本定理

ルービックキューブのすべての面が揃った状態で、

・2 面体 uf の上面側の小面

・2 面体 ur の上面側の小面

・2 面体 fr の前面側の小面

・スライス群(中間層を回転させる手順で生成される群)に含まれる手順によって上記の三面の小面が移るすべての小面

・それぞれの 3 面体の上面側及び下面側の小面

に印をつける。この印を基準参照印と呼ぶ。一方、ルービックキューブの手順gによって基準参照印の新しい配置が決まる。これをgに対する印と呼ぶ。

定理([1]) 次の決定過程によってルービックキューブの配置は決定される。

(a)2 面体がどのように置換されたか。

(b)3 面体がどのように置換されたか。

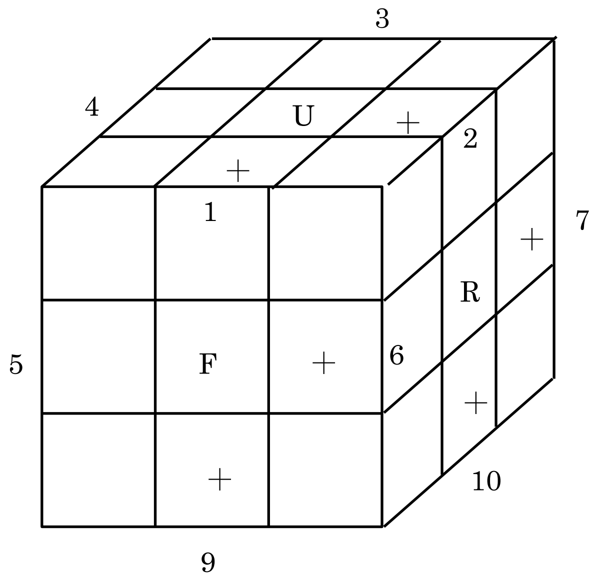
(c)どの 2 面体の印が(基準参照印に対して)反転したか

(d)どの 3 面体の印が(基準参照印に対して)どれだけ(時計回りに 120 度または 240 度)回転したか。

3.2 小方体の表記

3.2.1 2 面体の向き

キューブの第 1 基本定理にあるように 2 面体と 3 面体の向きを決める。前面(F)、右面(R)、上面(U)にある 2 面体の向きと番号付けは以下の図の通り。



残りの面は 2 面体の dl,db,lf,lu,bl,bu 面に印を付ける。これで 2 面体の向きを決める。辺の番号付けは辺 uf,ur,ub,ul,lf,fr,rb,bl,df,dr,db,dl の順に 1 から 12 の番号を振る。

$\mathrm{w}: \mathrm{H} \to C_2^{12}$をそれぞれの手順$g \in \mathrm{H}$に対して、その手順の結果における 2 面体の向きを対応させる関数とする。より正確にいうと、$g \in \mathrm{H}$が 2 面体iをjに移すとき$w_i(g) \in C_2$はgによって移されるi番目の 2 面体の向きである。ただし辺は図で表したように番号付けされてお

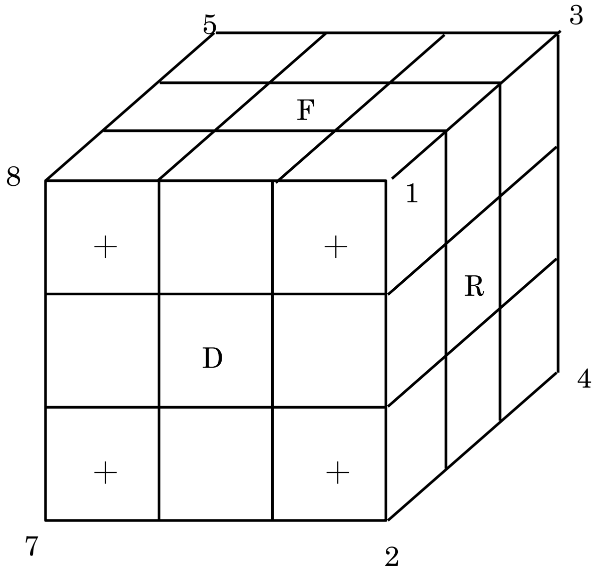
り 2 面体iが手順gによって辺jに動かされたときに 2 面体iの印のついた小面を辺jの印のついた小面に合わせるために反転させる角度が 180°の何倍かによって向きを表すものとする。

単位操作およびそれぞれの組み合わせによって 2 面体の向きは以下の表の通りになる。

X	$\vec{w}(X)$
F	(1,0,0,0,0,0,0,0,1,0,0,0)
U	(1,0,1,0,0,0,0,0,0,0,0,0)
F*U	(1,0,1,0,1,0,0,0,1,0,0,0)
U*F	(1,1,1,0,0,0,0,0,1,0,0,0)
B	(0,0,0,0,0,0,1,1,0,0,0,0)
D	(0,0,0,0,0,0,0,0,0,1,0,1)
R	(0,1,0,0,0,0,0,0,0,1,0,0)
L	(0,0,0,0,1,0,0,0,1,0,0,0)

3.2.2　3 面体の向き

下側の全ての小面と上側の全ての小面に印を付ける。前面(F)、右面(R)、下面(D)の 3 面体の向きおよび番号付けは以下の通り。



$v: \mathrm{H} \to C_3^8$をそれぞれの手順$g \in \mathrm{H}$に対して、その手順の結果における 3 面体の向きを対応させる関数とする。より正確にいうと、$g \in \mathrm{H}$が 3 面体iを頂点jに移すとき$v_i(g) \in C_3$はgによって移されるi番目の 3 面体の向きである。ただし頂点は図で表したように番号付けされていて 3 面体iが手順gによって頂点jに動かされたときに 3 面体iの印のついた小面を頂点j

の印のついた小面に合わせるために時計回りに捻る角度が 120° の何倍かによって向きを表すものとする。

単位操作によって 3 面体の向きは以下の表の通りになる。

X	$\vec{v}(X)$
F	(2,0,1,0,1,0,0,2)
U	(0,0,0,0,0,0,0,0)
D	(0,0,0,0,0,0,0,0)
B	(0,1,0,2,0,2,1,0)
R	(1,2,2,1,0,0,0,0)
L	(0,0,0,0,1,2,1,2)

それぞれの$g \in \mathrm{G}$を

$$\left(\vec{v}(g), \rho(g), \vec{w}(g), \sigma(g)\right)$$

と同一視する。ここで

・$\rho(g)$はgによるキューブの 3 面体の集合 V の置換とする。

・$\sigma(g)$はgによるキューブの 2 面体の集合 E の置換とする。

・$\vec{v}(g)$および$\vec{w}(g)$は 3 面体および 2 面体の向きとする。

3.3　キューブの第 2 基本定理

定理([1])　次の条件を満たすとき、そしてその時に限りルービックキューブは四つ組 $(\vec{v}, r, \vec{w}, s)(r \in S_8, s \in S_{12}, \vec{v} \in C_3^8, \vec{w} \in C_2^{12})$に対応する配置となることができる。

(a)$\mathrm{sgn}(r) = \mathrm{sgn}(s)$　(「置換の奇遇性の一致」)

(b)$v_1 + \cdots + v_8 \equiv 0(mod\ 3)$　(「総捻り量の保存」)

(c) $w_1 + \cdots + w_{12} \equiv 0(mod\ 2)$　(「総反転量の保存」)

ただし$\mathrm{sgn}(f)$は置換fの符号である。

証明

(a)

$g \in \mathrm{G}$をルービックキューブのすべての面が揃った配置からこの四つ組に対応する手順とする。すると$\mathrm{r} = \rho(g)$および$\mathrm{s} = \sigma(g)$となる。gは単位操作 R,L,U,D,F,B の語として表せるので$g = X_1 + \cdots + X_k$とする。ただし、それぞれのX_iは R,L,U,D,F,B のいずれかである。X を単位操作のいずれかとするとき$\mathrm{sgn}\left(\rho(X)\right) = \mathrm{sgn}\left(\sigma(X)\right)$が成り立つ。$\mathrm{sng}, \rho, \sigma$は準同型写像なので次の式が成り立つ。

$$\mathrm{sgn}(r) = \mathrm{sgn}\big(\rho(g)\big) = \prod_{i=1}^{k} \mathrm{sgn}\big(\rho(X_i)\big) = \prod_{i=1}^{k} \mathrm{sgn}\big(\sigma(X_i)\big) = \mathrm{sgn}\big(\sigma(X)\big) = \mathrm{sgn}(s)$$

(b)

単位操作について次が成り立つ。

(i)$\vec{v} = (\overrightarrow{v_1}, \cdots, \overrightarrow{v_8})$が総捻り量の保存条件を満たすのは、その成分の置換の一つ $\mathrm{P}(p)(\vec{v}) = \big(\overrightarrow{v_{(1)p}}, \cdots, \overrightarrow{v_{(8)p}}\big)$がその条件を満たす時に限る。

(ii) $(\overrightarrow{v_1}, \cdots, \overrightarrow{v_8})$および$\big(\overrightarrow{v'_1}, \cdots, \overrightarrow{v'_8}\big)$がそれぞれ総捻り量の保存条件を満たすとき、この二つの和もこの条件を満たす。

上記と同じようにgを単位操作 R,L,U,D,F,B の語$g = X_1 \cdots X_k$として表す。それぞれのX_iは単位操作 R,L,U,D,F,B のいずれかである。ただしこの式はkができるだけ小さくなるようにX_iを選んだという意味で最短の式になっていると仮定する。このkをgの長さという。

$k = 1$のとき、(b)は成り立つ。

$k > 1$として、二つの手順の積の向きはそれぞれの手順の向きを使って次のように表せる。

$$\vec{v}(X_1 \cdots X_{k-1} X_k) = \rho(X_1 \cdots X_{k-1})^{-1}\big(\vec{v}(X_k)\big) + \vec{v}(X_1 \cdots X_{k-1})$$

項$\rho(X_1 \cdots X_{k-1})^{-1}\big(\vec{v}(X_k)\big)$は(i)より総捻り量保存条件を満たす。項$\vec{v}(X_1 \cdots X_{k-1})$は帰納法の仮定より総捻り量の保存条件を満たす。よって(ii)より二つの和は総捻り量の保存条件を満たす。これで(b)が証明できた。

(c) ((c)の証明は筆者が考察した)

2 面体の向きの表より(c)と次が成り立つ。

(i)$\vec{w} = (w_1 \cdots w_{12})$が(c)の総反転量の保存条件を満たすのは、その成分の置換の一つ$P(p)(\vec{w}) = \big(w_{(1)p} \cdots w_{(12)p}\big)$がその条件を満たす時に限る。

(ii) $(w_1 \cdots w_{12})$および$(w'_1 \cdots w'_{12})$がそれぞれ(c)の総反転量の保存条件を満たすとき、この二つの和もこの条件を満たす。

上記と同じようにgを単位操作 R,L,U,D,F,B の語$g = X_1 \cdots X_k$として表す。それぞれのX_iは単位操作 R,L,U,D,F,B のいずれかである。ただしこの式はkができるだけ小さくなるようにX_iを選んだという意味で最短の式になっていると仮定する。

$k = 1$のとき、(c)は成り立つ。

$k > 1$として、二つの手順の積の向きはそれぞれの手順の向きを使って次のように表せる。

$$\vec{w}\big((X_1 \cdots X_{k-1})X_k\big) = \sigma(X_1 \cdots X_{k-1})^{-1}\big(\vec{w}(X_k)\big) + \vec{w}(X_1 \cdots X_{k-1})$$

X_kは単位操作であるから$\vec{w}(X_k)$は(c)の総反転量の保存条件を満たす。よって項$\sigma(X_1 \cdots X_{k-1})^{-1}\big(\vec{w}(X_k)\big)$は(i)より総反転量の保存条件を満たす。項$\vec{w}(X_1 \cdots X_{k-1})$は帰納法の仮定より総反転量の保存条件を満たす。よって(ii)より二つの和は総反転量の保存条件を満

たす。これで(c)が証明できた。

以上の(a),(b),(c)よりキューブ理論の第 2 基本定理が証明できた。

第 4 章　ルービックキューブの解法

4.1　神のアルゴリズム

ルービックキューブをあらゆる状態から解くには最短で何手必要かという問題を多くの人が解こうとしてきた。その最短手数は神のアルゴリズムと呼ばれ様々な研究がなされてきた。2010 年に米 Google の支援を受けた国際チームがルービックキューブの全パターンを調べ上げ、どんな状態からでも最短 20 手で解くことができると突き止めた。米 Google は高性能コンピューターを用いて数週間かけて突き止めた。ただしこれは最短の手数が 20 手だと突き止めただけであって、神のアルゴリズムを突き止めたわけではない。ルービックキューブを解いた手順からアルゴリズムを見つけることはできなかった。

神のアルゴリズム自体はまだ見つかっていない。

4.2　ルービックキューブを解くアルゴリズム

SAGE において実装されているルービックキューブの解法アルゴリズム[1]は以下の通り。

(1)6 つの面に対する手の生成元による自由群を構成し、生成元が満たすべき関係の適切な集合を計算する。

(2)ルービックキューブ群からこの自由群の剰余群への準同型を計算する。

(3)ルービックキューブの配置を群の元とみて、それを自由群の商に写像する。

(4)生成元の関係及び組み合わせ群論の手法を使って、この写像された元に対する「語の問題」を解く。

(5)Python の文字列パース関数を使って、これをルービックキューブの表記に書き換える。

4.3　プログラム

以下のプログラムは SAGE に実装されているルービックキューブの解法プログラムである。

ルービックキューブの状態を置換群の元として入力している。キューブの配置を変化させ、それを SAGE に揃えさせている。以下のプログラムでは状況を単純にするために $R*U*R^{-1}$ のような単純な手順でキューブの配置を変えている。

```
sage: rubik = CubeGroup()
sage: b = rubik.B()
sage: d = rubik.D()
```

```
sage: f = rubik.F()
sage: l = rubik.L()
sage: r = rubik.R()
sage: u = rubik.U()
sage: r*u*r^(-1)
        (1,19,24,6)(2,21,7,4)(8,30,17,9)(10,34,28,18)(11,35,25,43)
sage: G = PermutationGroup([b,d,f,l,r,u])
sage: g = G(str(r*u*r^(-1)))
sage: g.word_problem([b,d,f,l,r,u])
        [['(3,38,43,19)(5,36,45,21)(8,33,48,24)(25,27,32,30)(26,29,31,28)',        1],
        ['(1,3,8,6)(2,5,7,4)(9,33,25,17)(10,34,26,18)(11,35,27,19)',               1],
        ['(3,38,43,19)(5,36,45,21)(8,33,48,24)(25,27,32,30)(26,29,31,28)', -1]]
        ('x5*x6*x5^-1', (3,38,43,19)(5,36,45,21)(8,33,48,24)(25,27,32,30)(26,29,31,28)
        *(1,3,8,6)(2,5,7,4)(9,33,25,17)(10,34,26,18)(11,35,27,19)*(3,38,43,19)(5,36,45,21)
        (8,33,48,24)(25,27,32,30)(26,29,31,28)^-1')
```

解はword_problemコマンドの出力であるx5*x6*x5^-1の次の行以降に出力されている。SAGEは生成元b,d,f,l,r,uをx1,x2,x3,x4,x5,x6に書き換えていることに注意しながら見ると、SAGEがキューブの配置を戻すのに手順rur^{-1}の逆を使っていることがわかる。

word_problemコマンドは任意の置換群に使うことができるが、ルービックキューブ群に対しては出力がより見やすく特化されたsolveコマンドが存在する。それを使ったプログラムは以下の通り。

```
sage: rubik = CubeGroup()
sage: state = rubik.faces("R*U")
sage: rubik.solve(state)
        'R U'
sage: state = rubik.faces("R*U*R^(-1)")
sage: state
        {[[43, 26, 27], [18, 0, 29], [17, 31, 32]], 'up': [[19, 21, 3], [2, 0, 5], [1, 4, 30]], 'back':
        [[33, 28, 25], [36, 0, 37], [38, 39, 40]], 'down': [[41, 42, 11], [44, 0, 45],
          [46, 47, 48]], 'front': [[9, 10, 24], [20, 0, 7], [22, 23, 6]], 'left': [[8, 34, 35],
          [12, 0, 13], [14, 15, 16]]
sage: rubik.solve(state)
        'R U R'
```

4.4　人の手による解法

人の手でルービックキューブを解く場合は様々なやり方が考案されている。その中で基本的な解き方であるコーナー/エッジ法とレイヤー法についてと、スピードキューブの大会で最もポピュラーな解法である LBL 法について紹介する。

4.4.1　コーナー/エッジ法

(1)まず小方体の向き、すなわち捻りや反転を無視して 3 面体の位置を揃える。
これは、3 面体を捻りが加えられているのを無視して 1 面体の色に合わせるということである。
(2)次に小方体の向き、すなわち捻りや反転を無視して 2 面体の位置を揃える。
これは 2 面体を反転が加えられているのは無視して 1 面体の色に合わせるということである。
(3)3 面体の向きを揃える。つまり、3 面体を捻る操作を行う。これで全ての 3 面体は完全に揃う。
(4)2 面体の向きを揃える。つまり、2 面体を反転させる操作を行う。これで完成である。

4.4.2　レイヤー法

(1)上層(上面)にある 3 面体および 2 面体を揃える。
(2)中層にある 2 面体を揃える。(できる限り下層にある 2 面体も揃える。)
(3)下層(下面)にある 3 面体を(必要であれば 2 面体も)揃える。

4.4.3　Layer By Layer(略称 LBL 法)

(1)上面を十字に揃える。この手は一般に CROSS と呼ばれている。
(2)ルービックキューブを逆さにし、First Two Layer(略称 F2L 法)を用いて下層、中層を揃える。基本パターンは 41 パターン存在する。
(3)Orienting Last Layer(略称 OLL 法)を用いて捻りや反転を無視して上面を揃える。基本パターンは 57 パターン存在する。
(4)Permuting Last Layer(略称 PLL 法)を用いて上層側面を揃える。基本パターンは 21 パターン存在する。

4.5　実験

4.5.1　概要

SAGE を使い、コンピューターを用いて求める解の手数と、LBL 法を用いて人の手によって求める解の手数を比べる。

任意の手数でキューブの配置をランダムに変えて、それを元の状態に戻す手数を数える。

キューブの配置を変える手順を同じにすることで、同じ状態からコンピューターと人の手で戻すことができる。キューブの配置に間違いがないように SAGE で rubiksCube().move("").show3d()コマンドを用いてコンピューター上でのキューブの配置を表示させて確認している。また、rubik.plot_cube コマンドを使って展開図も表示させて確認を行った。手順は時計回り、もしくは反時計回りに 90° 回転させる操作を一回と数える。R, L, U, D, F, Bは時計回り、R^-1,L^-1,U^-1,D^-1,F^-1,B^-1 は反時計回りの操作とする。

　キューブの配置を変える手数は 30,40,50,60,70,80,90,100 の回数である。

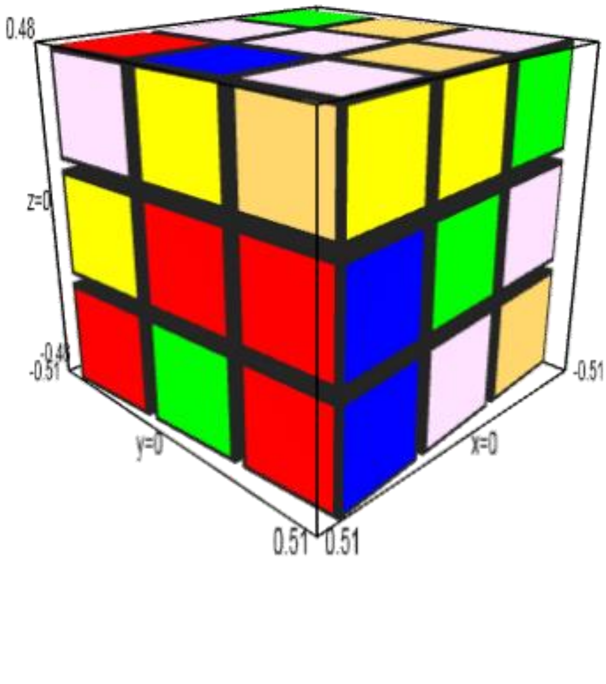
4.5.2　結果

(1)30 手回転させた場合

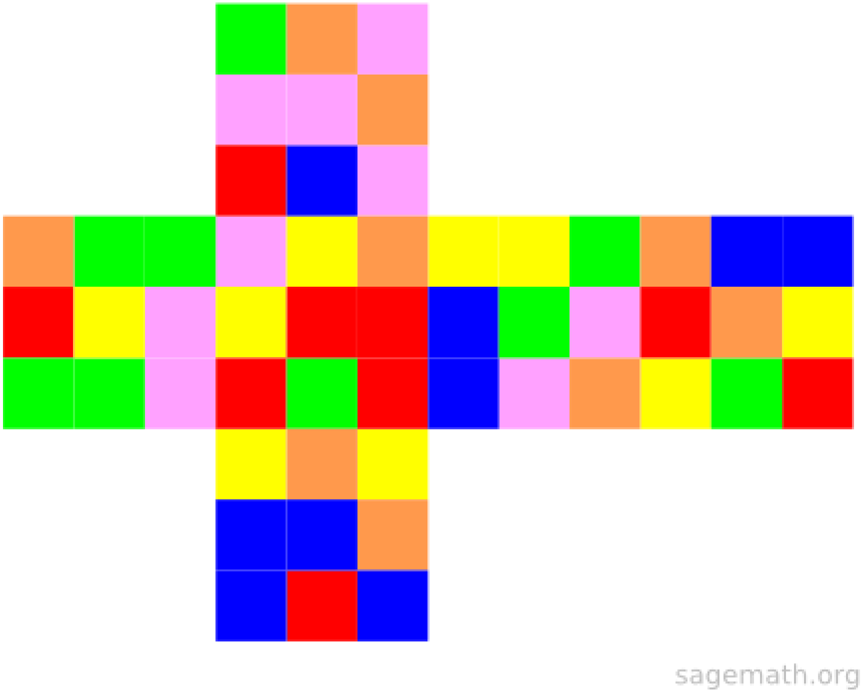
回転手順

R^-1*L^-1*R^-1*D^-1*B^-1*F^-1*U^-1*D^-1*U^-1*F*D*B^-1*D*L*U*B^-1*F^-1*B^-1*D*L*L*B^-1*U*D^-1*D^-1*L*B*U^-1*D*B

回転結果



展開図



コンピューターによる解法

D^-2*R^-1*B^-2*U^-2*F*B^-2*L^-1*U*R^-1*F^-2*D^-1*F*R^-2*L^-2*U^-2*F*B^-1*
L^-2*B^-1

28 手

LBL 法

F*L*L*U^-1*L^-1*B*B*D*L*B^-1*L^-1*U*U*B*D^-1*B*D*R*D^-1*R^-1*B*B*L*D*
L^-1*D*R*D^-1*D^-1*R^-1*D*R*D^-1*R^-1*D^-1*D*R^-1*D*D*R*D^-1*D^-1*R^-1*
D*R*D^-1*D*L^-1*D*F*D^-1*D^-1*F^-1*D*F*F*D*F^-1*L^-1*F*L*D^-1*L^-1*F^-1
*L*B*D^-1*F*D^-1*D^-1*B^-1*D*F^-1*B*D^-1*F*D^-1*D^-1*B^-1*D*F^-1*D

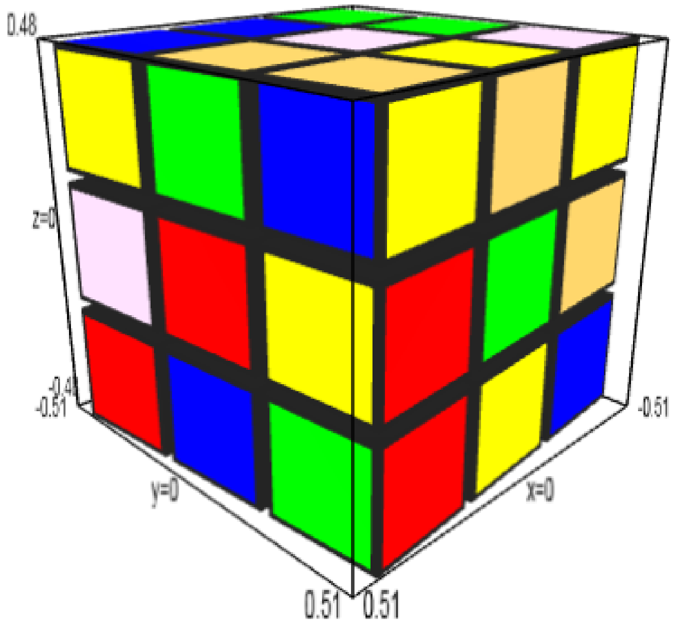
82 手

(2)40 手回転させた場合

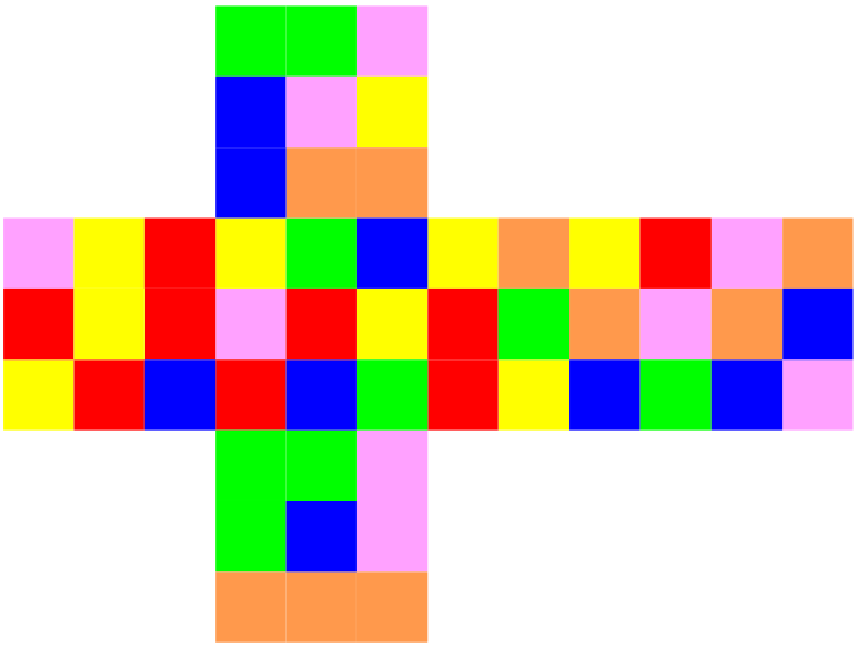
回転手順

B*L-1*D*L*U^-1*B^-1*D*B*D^-1*U^-1*D^-1*B*U*L*D^-1*L*U^-1*B^-1*L^-1*D
F*B^-1*D^-1*L*D*L*L*B^-1*B^-1*U*U^-1*B*U-1*D*D*L*U*B*R*R

回転結果



展開図

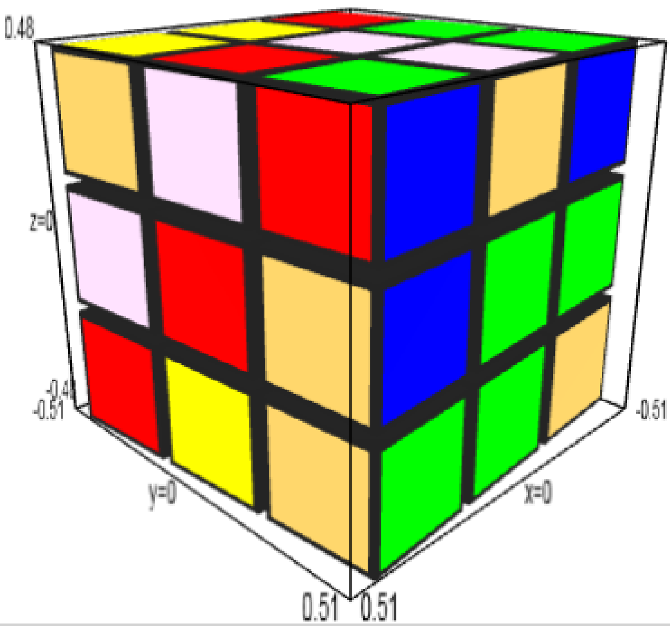


コンピューターによる解法
R^-1*F^-1*D^-2*B*L*D^-1*B^-2*U^-1*R^-1*B^-1*U^-1*L^-1*F^-2*U^-2*R^-2*F^-1*D^-2*B^-1*R^-2*U^-2
28 手

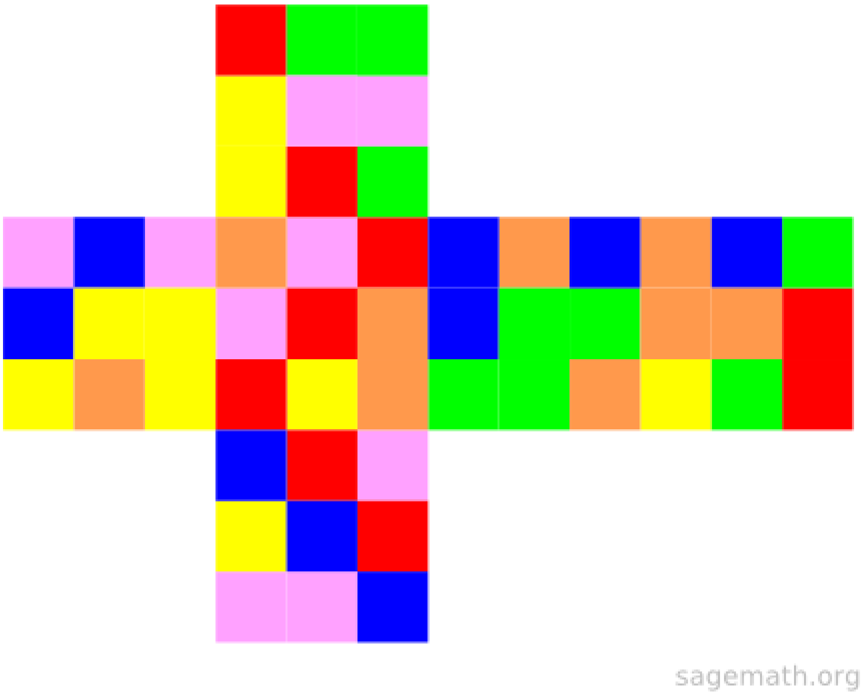
LBL 法
L^-1*D*R^-1*B*B*U^-1*B*R^-1*B^-1*D*D*F*D*F^-1*D^-1*D^-1*R*D^-1*R^-1*D*B^-1*D^-1*B*L*D*D*L^-1*F*D*F*L^-1*F^-1*L*D*D*F^-1*D*B^-1*L*B*L^-1*D*B*D*B^-1*D*D*L*D^-1*L^-1*F^-1*D*D*F*L^-1*F^-1*R*F^-1*R^-1*F*F*L*L^-1*D*R*D^-1*D^-1*L*D^-1*R*D^-1*D^-1*B*F*D^-1*D^-1*B^-1*F^-1
78 手

(3)50 手回転させた場合
回転手順
L*B*L*L*U^-1*D*B*R*R*U*L^-1*D*B^-1*L^-1*D^-1*U*U*D^-1*L*R^-1*U*B*L*R*R*D^-1*R*U*L^-1*D*D*R^-1*L^-1*B^-1*B^-1*L*D^-1*D^-1*B*U*R^-1*R^-1*L*L*B*D^-1*L^-1*L^-1*U*U

回転結果



展開図



コンピューターによる解法

D^-1*B^-2*D^-2*B^-2*U^-1*F^-1*R^-2*L^-1*F^-1*R*L^-1*U^-2*L^-2*B^-2*R^-2*L^-2*B*U^-2

30 手

LBL 法

U^-1*L^-1*D^-1*R^-1*F^-1*R^-1*D^-1*F^-1*D*F*D^-1*D^-1*L*D*L^-1*D^-1*D^-1*B*D^-1*D^-1*B^-1*D^-1*L^-1*D^-1*L*D^-1*F*D*F^-1*D*D*F*D^-1*F^-1*D^-1*D*B^-1*D^-1*B*D^-1*D^-1*B^-1*D*B*R^-1*D^-1*B^-1*D^-1*B*D^-1*B^-1*D^-1*B*R*L^-1*R^-1*D*D*L*R*B*F*D*D*B^-1*F^-1

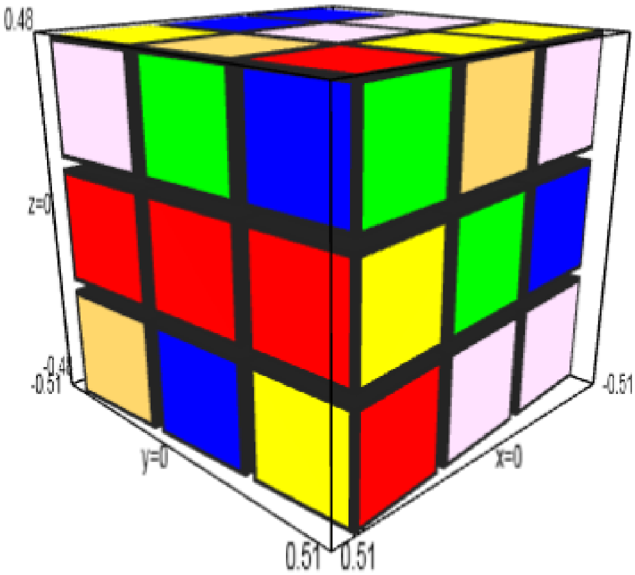
66 手

(4)60 手回転させた場合

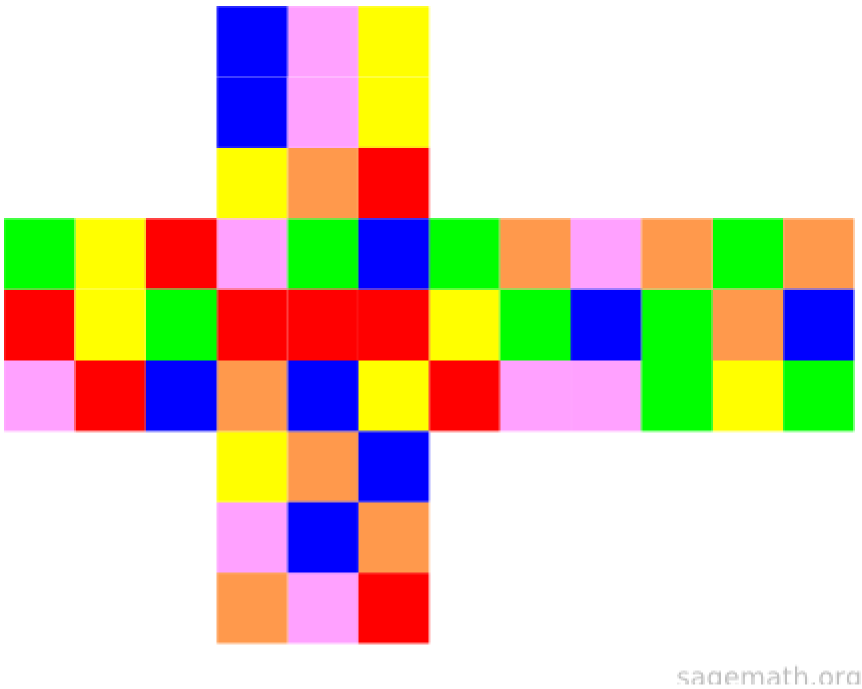
回転手順

R*L*B^-1*L*L*U^-1*D*F*F*U*D*L^-1*L^-1*B^-1*R*R*D*F^-1*U^-1*R^-1*U*L*L*D*F*B*B*F^-1*F^-1*D*B*F^-1*U^-1*U^-1*L*L*B*D*F*R^-1*U^-1*L*U^-1*D^-1*B*L*R*R*D^-1*F*B^-1*U*U*R*L^-1*B*F^-1*U*B^-1*D^-1

回転結果



展開図

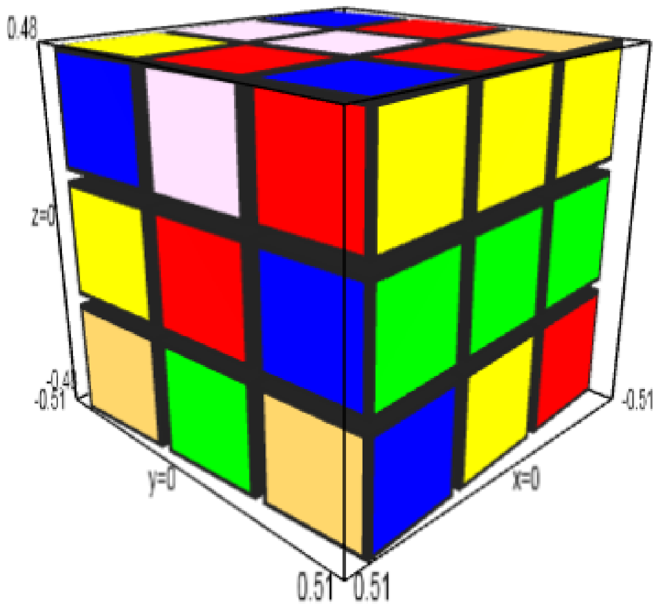


コンピューターによる解法
R^-2*L^-2*B*D^-1*U^-2*B*L*B^-2*L^-2*F^-1*D*U*F^-2*B^-1*L^-2*F^-2*B^-1*D^-2*B
28 手

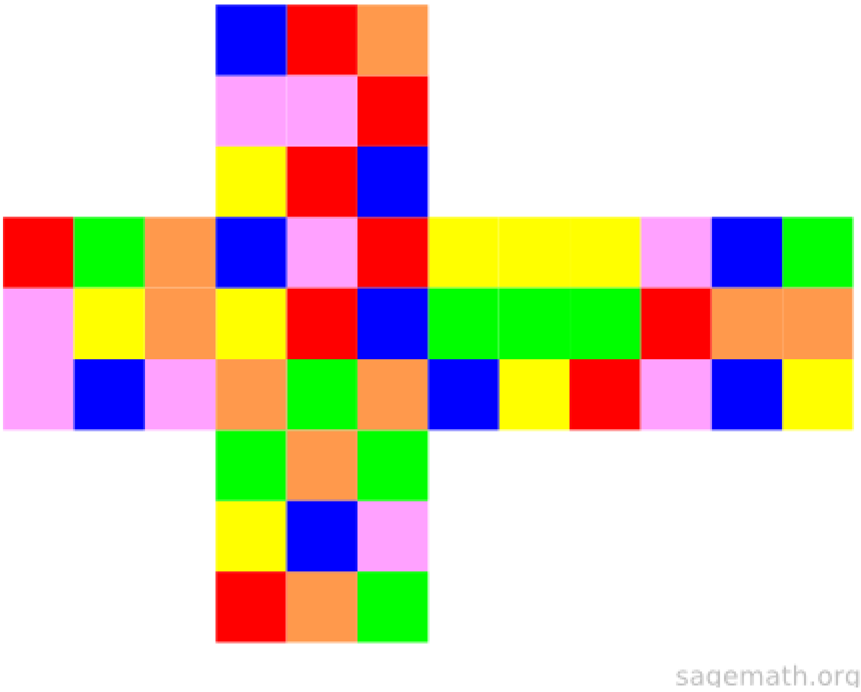
LBL 法
U^-1*R*R*D^-1*B*B*L^-1*F*L*U*U*D^-1*L^-1*D*L*D^-1*D^-1*B*D*B^-1*L*D*L^-1*D^-1*D*R^-1*D*D*R*D^-1*D^-1*R^-1*D*R*D^-1*B^-1*D^-1*B*F^-1*L*D*D*L^-1*D^-1*D^-1*L^-1*F*F*L*F^-1*D^-1*F*F*D^-1*R*L^-1*F*F*L*R^-1*D^-1*F*F
63 手

(5)70 手回転させた場合
回転手順
D*D*L^-1*B^-1*F*F*B*R^-1*D*U*D*U^-1*B*B*L^-1*L^-1*R*U*R*U*F^-1*B*L*U^-1*U^-1*R^-1*R^-1*R^-1*B*L*F*F*B^-1*D^-1*R*L*U^-1*B*-1*F^-1*B^-1*L*L*U*R^-1*F^-1*F^-1*B*R*D^-1*D^-1*F*F*B^-1*R^-1*U^-1*U^-1*R^-1*B^-1*F*D*U*L*R*R*B^-1*F^-1*U^-1*D*D*B^-1

回転結果



展開図



コンピューターによる解法

D*B^-1*L*F^-2*U^-2*B^-1*R^-2*D*R^-1*B^-1*L^-1*B*U^-2*R^-2*F^-2*B^-1*U^-2*F*L^-2*B

28 手

LBL 法

R*R*U*U*B^-1*U^-1*R^-1*U*D*B^-1*D*D*B*D^-1*R*D*R^-1*D^-1*D*R^-1*D*R*D^-1*D^-1*R^-1*D*R*B*D*B^-1*D*F^-1*D*F*D^-1*F^-1*D^-1*F*D^-1*L^-1*D*D*L*D^-1*L^-1*D*L*F^-1*B*B*R*B^-1*R*F*D*D*F^-1*R*F*B^-1*L*R^-1*D*D*L*R*B*D^-1*F*D*D*B^-1*D*F^-1*D*D

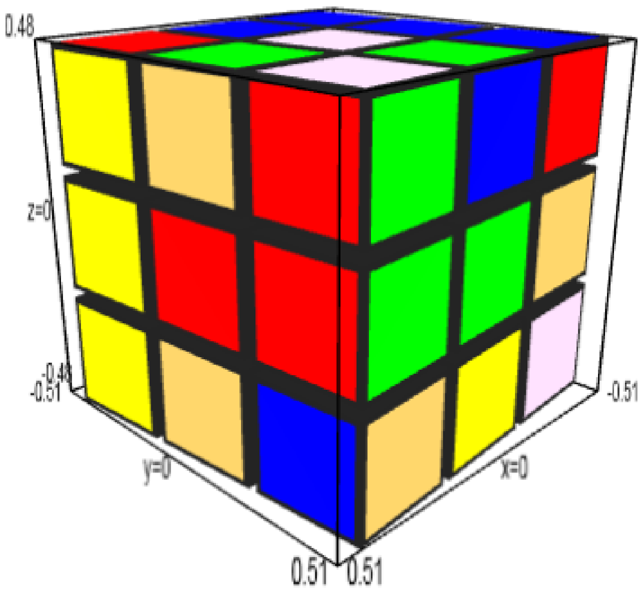
76 手

(6)80 手回転させた場合

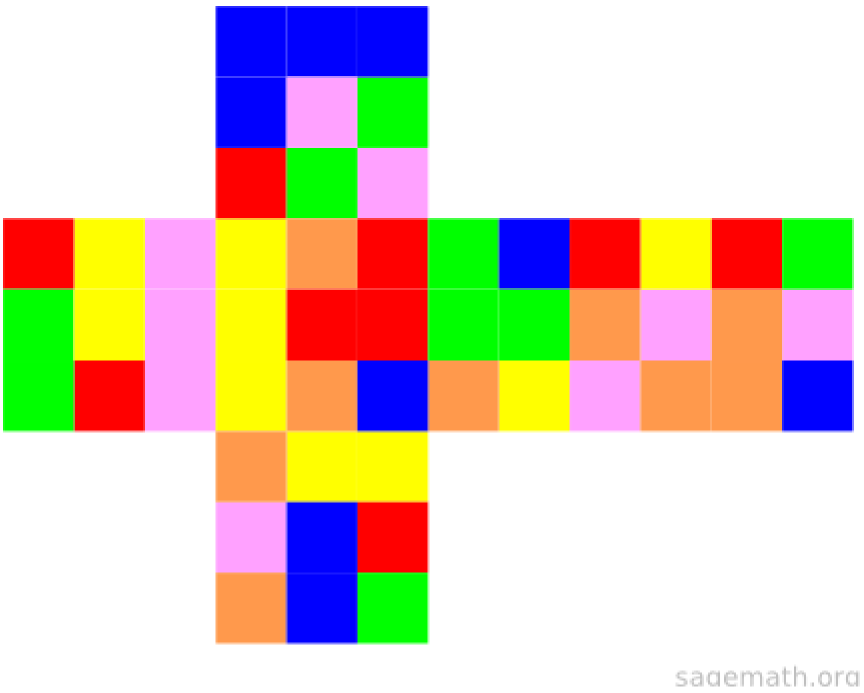
回転手順

D*L*U^-1*D*B*U^-1*F*F*L*R^-1*L*U^-1*U^-1*B*F*D*U*U*R*F*U^-1*L*B^-1*F*D*U^-1*R*R*U^-1*B*U^-1*L^-1*L^-1*L^-1*B^-1*F*F*D^-1*R*U^-1*R*F*F*D^-1*B*F^-1*D*B^-1*U~-1*U^-1*R^-1*R^-1*D*B^-1*U^-1*U^-1*B*^-1*U*U*R*B*B*L^-1*L^-1*U^-1*R*B*F*F*D^-1*L^-1*B*F*U*U*D*F*B^-1*F

回転結果



展開図



コンピューターによる解法

F*R*L*F^-2*U^-2*R^-2*L^-2*F^-1*U^-2*F^-1*D^-2*R*U^-1*F^-2*L^-2*D^-2*U^-2*B^-2

29 手

LBL 法

F*U^-1*L*U^-1*L*R^-1*U^-1*D^-1*F*D*F^-1*R*D*D*R^-1*D^-1*F^-1*D*D*F*D*F^-1*D^-1*F*L*D*L^-1*D^-1*F^-1*D^-1*F^-1*R*F*R^-1*D^-1*D^-1*F*B*D*B^-1*D*D*D^-1*R*D*R^-1*D*R*D*R^-1*D^-1*B*D^-1*B^-1*D*B*D*B^-1*L^-1*D^-1*L*D^-1*F*L*B^-1*L^-1*B*F^-1*D*B*R^-1*B^-1*R*L*B^-1*L*R^-1*B^-1*R*B*L*L*D

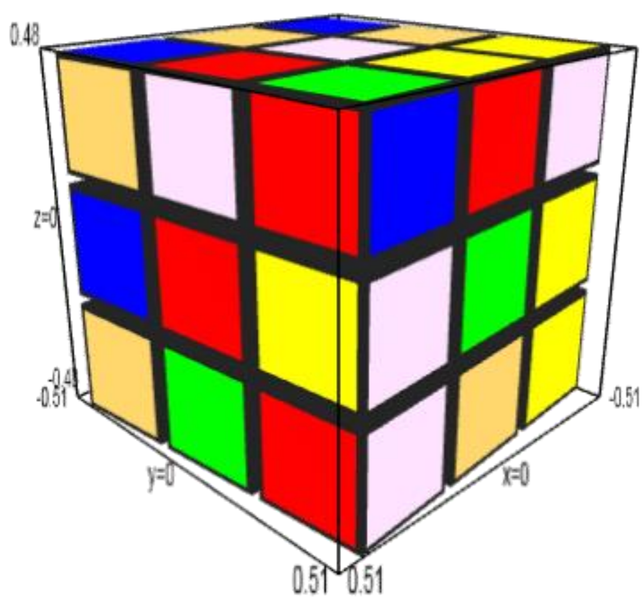
83 手

(7)90 手回転させた場合

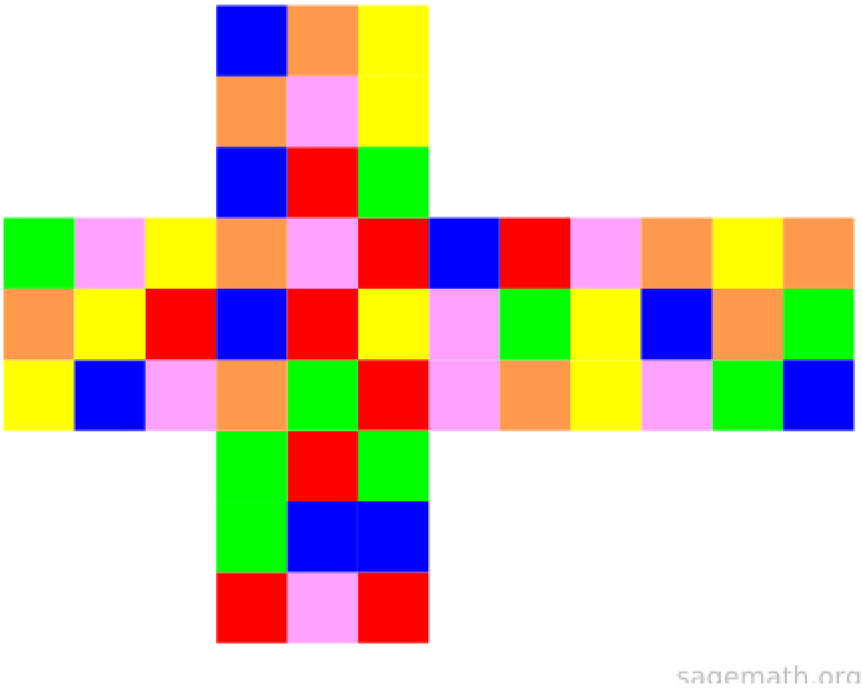
回転手順

R*F*D*D*F*B*L*U*U*L*F*B^-1*R^-1*R^-1*U*B^-1*F*D^-1*L*U*B^-1*U^-1*U^-1*F*D^-1*R^-1*B*L*R^-1*U^-1*L*B*B*F^-1*F^-1*D^-1*U*B^-1*U^-1*U^-1*R*B^-1*D*D*L^-1*B^-1*R*R*U*L^-1*U^-1*L^-1B*B*F*D^-1*U*R^-1*R^-1*B*F*R^-1*U*U*L^-1*L^-1*B*B*U^-1*R*U*F*F*D^-1*B^-1*R*D^-1*B*L*R^-1*U^-1*D*F^-1*F^-1*B^-1*U^-1*R*R*B*L^-1

回転結果



展開図



コンピューターによる解法

R^-2*D^-1*U^-1*L*U*B^-2*D^-2*B*D^-1*F^-2*R^-2*D^-1*L*D^-2*B*D^-2*F^-2*U^
-2*R^-2*B

30 手

LBL 法

F^-1*B*B*L*U^-1*F*U^-1*L*L*U^-1*D^-1*R^-1*D*D*R*D*R^-1*D^-1*R*D^-1*D^-
1*F^-1*D*D*F*D^-1*D^-1*F^-1*D*F*D^-1*D^-1*F^-1*D*F*D*B*D*B^-1*D^-1*D^-1
*L^-1*D*L*D^-1*R*D^-1*R^-1*D^-1*D^-1*B^-1*D*D*B*D^-1*B^-1*D*B*R^-1*B^-1
*D^-1*B*D*R*L*L*D^-1*F*B^-1*L*L*F^-1*B*D^-1*L*L

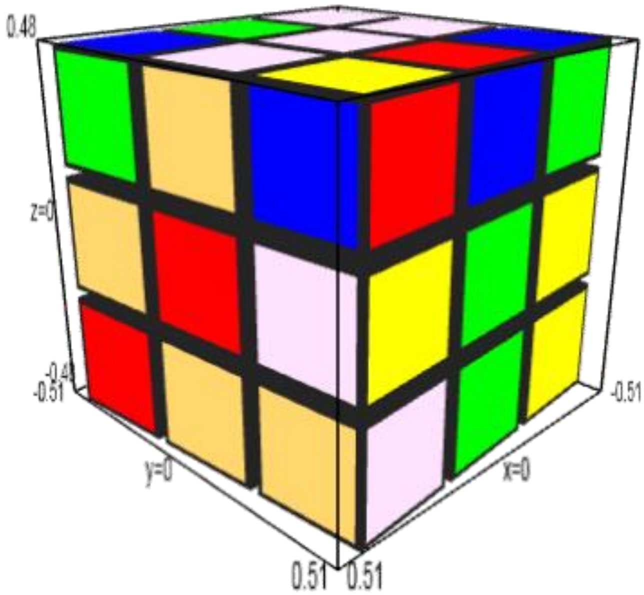
76 手

(8)100 手回転させた場合

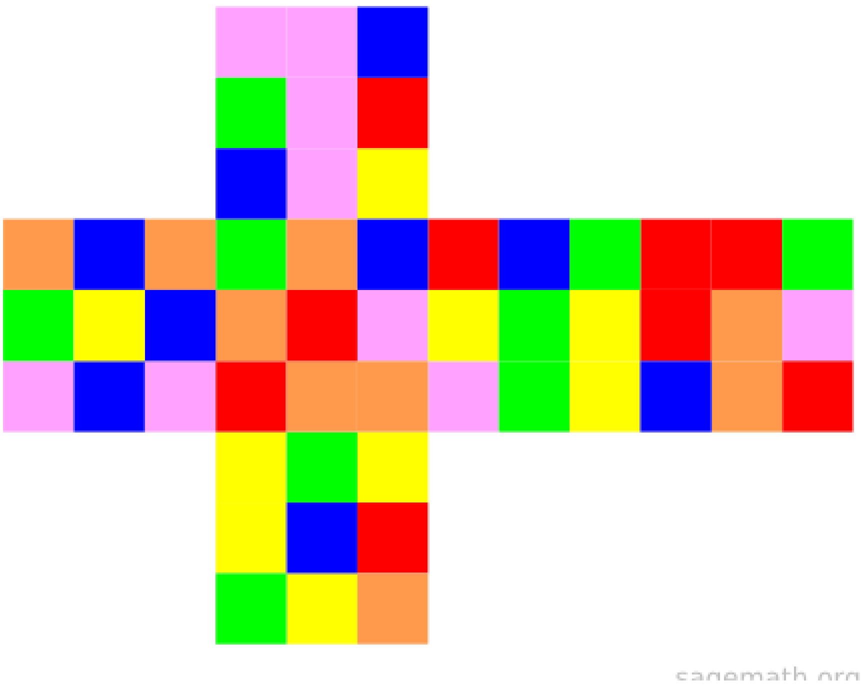
回転手順

L*L*B^-1*D^-1*F*R*R*U^-1*B*D*R^-1*L*B*U^-1*U^-1*R*F*F*D^-1*B*D*D*F^-1*
R^-1*R^-1*U*U*L^-1*L^-1*B*U*U*L*L*B*B*R*R*U*R*F^-1*D^-1*F^-1*U^-1*R^-1
*B^-1*L^-1*B^-1*F^-1*U^-1*R*L*B^-1*B^-1*D*U^-1*U^-1*B^-1*L*F^-1*D*F^-1*B
^-1*B^-1＊R*L*R*U*U*B*U^-1*D^-1*R*B*F^-1*D^-1*F^-1*R*L*B*R^-1*L*L*B^-1
*D*D*F^-1*F^-1*R*U*B^-1*D*F*R^-1*B*L^-1*U*L*R^-1*F

回転結果



展開図



コンピューターによる解法
R^-2*U*R^-1*F^-2*D^-2*L^-1*D*B^-1*L^-2*B^-2*D*L^-1*D^-2*F*D^-2*L^-2*U^-2*R^-2*F^-2
30 手

LBL 法
L*R*U*U*D*D^-1*L*D*L^-1*D*D*L*D^-1*L^-1*D*B*D*B^-1*D^-1*B*D*B*R^-1*B^-1*R*D*D*B^-1*F*D*F^-1*D^-1*F*D^-1*D^-1*F^-1*D^-1*F*D*F^-1*R*D*D*R^-1*D^-1*D^-1*D*R*D^-1*D^-1*R^-1*D*R*D^-1*R^-1*F*B*B*L^-1*B*L^-1*F^-1*D*D*F*L^-1*F^-1*B*F*L^-1*F*R*R*F^-1*L*F*R*R*F*F
80 手

以上の結果を表にまとめると以下の通りになる。

4.5.3　表

回数	コンピューター	人
30	28	82
40	28	76
50	30	66
60	28	63
70	28	76
80	29	83
90	30	76
100	30	80

第 5 章　考察・感想

実験結果からコンピューターの手数の平均は約 29 手、LBL 法の手数の平均は約 75 手であった。コンピューターではルービックキューブの回転数を変えても元に戻すまでの手数にはあまり変化はない。アルゴリズムが決まっていると手数も決まってくるのではないかと考察できる。LBL 法では最初の CROSS の手順では決まったパターンというものはないが F2L 法、OLL 法、PLL 法では決まったパターンが存在する。上手く最短のパターンが続いたとしても約 40 手必要である。平均 29 手であるコンピューターには及ばない。手数を減らすにはアルゴリズム自体を変えなければならないと思う。どういった手順でルービックキューブを初期状態に戻しているのかを確認するために LBL 法の手順をみると、交換

子や共役を多数用いて順番に揃えているのがわかる。手順を少なくするために福岡教育大学の藤本光史教授の手順を簡約させるプログラム[2]を用いてみた。今回の実験のうち 100 手ルービックキューブを回転させた状態のルービックキューブを LBL 法で得られた解法にそのプログラムを用いると、R^-1*F*R*F*U*R^-1*L^-1*F*F*U*F*U*R^-1*B*L*D^-1*F*B*B*R^-1 という結果を得られた。20 手まで減ったが、使われているアルゴリズムとしては不明である。この結果を含めて、コンピューターの解法ではどういった順序で揃えているのかがわからない。下層、中層、上層の順番に揃えていく方法ではないのでアルゴリズムとして認識しづらい。「人間にはわからず神のみが認識できるアルゴリズム」が語源である、神のアルゴリズムの意味を感覚的に知ることができた。

LBL 法はスピードキューブの大会でよく使われている方法なので、揃える速さはどうなのか調べてみた。世界記録は約 5 秒であった。40 手で完成したとしても、1 秒につき 8 手回転させた計算になる。それに対してコンピューターを用いて機械に揃えさせた記録は約 1 秒であった。人の認識できるアルゴリズムでは速さにも限界があるのだろう。

今回、群論とルービックキューブについて研究したが、まだ基本的な部分でしかない。アダマール行列やケーリーグラフなど解析方法は多い。今後の展開としてはさらに深く群論を使って解析することや、群論以外での解析について調べることが挙げられる。また、SAGE では回転手順を入力して初期状態に戻すプログラムであったが、変化した後の状態を入力して、それを初期状態に戻すプログラムの作成を行うことも今後の課題として挙げられる。

第 6 章　謝辞

本研究を進めるに当たり、ご教授を頂いた卒業論文指導員の白柳潔教授に感謝致します。

第 7 章　参考文献

[1]David Joyner 著(川辺治之訳)　『群論の味わい　～置換群で解き明かすルービックキューブと 15 パズル～』　共立出版株式会社　2010 年 12 月 10 日発行

[2]藤本光史　『数式処理を用いたルービックキューブの素数位数操作の探求』 http://catalog.lib.kyushu-u.ac.jp/handle/2324/1430848/p069.pdf　2009 年