Two Two-Cycles

Two 2-cycles are particular permutations of cube pieces where 4 pieces are permuted 2 by 2. Two 2-cycles are even-parity permutations. They can be represented as the composition of two transpositions (i j) (k l).

From group theory, it is known that any permutation of a group can be expressed as the composition (product) of transpositions – formally, they are *generators* for the group. In an *alternating* group, all permutations are of *even* parity and thus can be expressed as the composition of an *even* number of transpositions. They can be further grouped 2 by 2 to give a set of two 2-cycles only, without any single transposition left aside. This shows that *any* member of an alternating group can be generated by using two 2-cycles only, hence the importance of generating a *complete* database of (possibly short) algorithms of two 2-cycles.

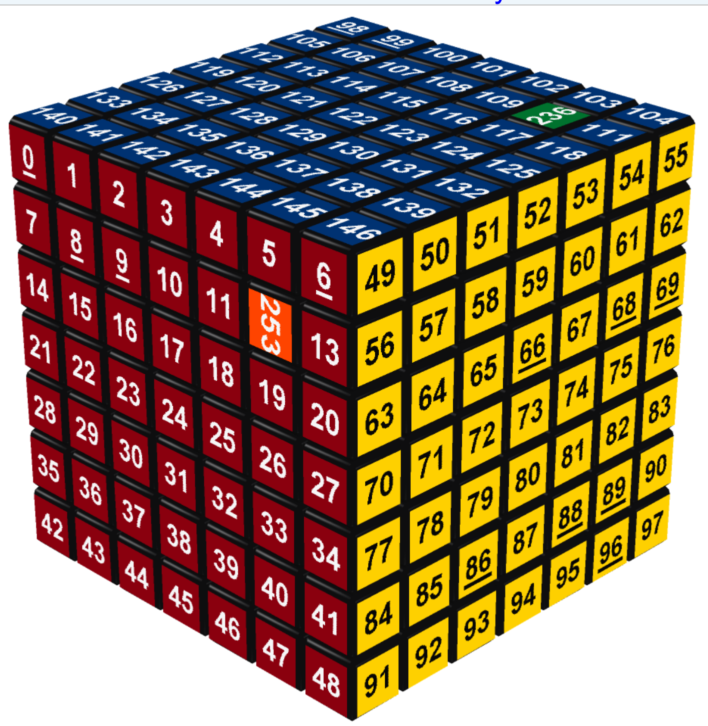
Short algorithms of two 2-cycles of pieces can be found by running algorithm templates in the VBA version of Algorithm Finder to give *seeds*. A seed algorithm is defined as a *short* algorithm that could generate a set of many other algorithms, by rotation/reflection, inversion, cyclic-shifting, or by adding setup moves. Seed algorithms are *irreducible*, meaning that they can't be obtained from each other by inversion, cyclic-shifting or symmetry considerations. See http://www.mementoslangues.fr/CubeDesign/CubeTheory/SeedAlgorithms.pdf for more information.

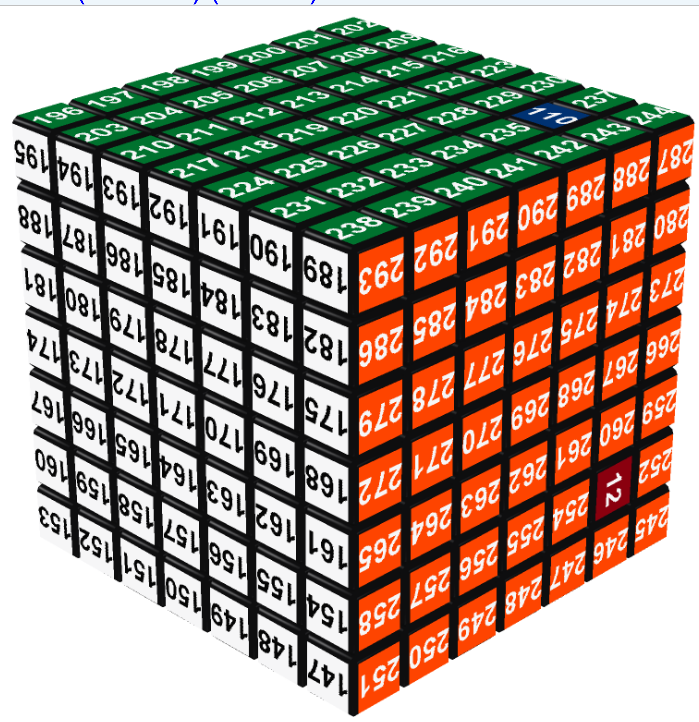
Seed algorithms are sorted out in stacks, where the first selected items are the shortest algorithms found. Each selected seed algorithm in the stack is then transformed by rotation/reflection, inversion and cycle shifting. Setup moves may be further added before algorithm execution and comparison of the new cube state with a goal state. If the two states are identical, then the transformed algorithm is added to the database. This is continued until the database is complete. Stacks of *scalable* seed algorithms have been generated for two 2-cycles of:

- Corners
- Midges
- Edges
- Corner-Centers
- Midge-Centers
- Edge-Centers

Scalable Seed Algorithm Example – Corner-Centers – Two 2-Cycles

Two 2-Cycles of Corner-Centers (110 236) (12 253)





NR' NU' R' L NU' NR NU R L' NU = [NR' NU' R' L, NU'[

Basic Group Theory

Group Theory – Useful Links
http://en.wikipedia.org/wiki/Permutation
http://en.wikipedia.org/wiki/Symmetric_group
http://en.wikipedia.org/wiki/Conjugacy_class
http://groupprops.subwiki.org/wiki/Cycle_type_of_a_permutation
http://en.wikipedia.org/wiki/Coset

By applying basic group theory to the cube, it can be inferred that (see links above for more details):

Conjugacy Class:

If A is an algorithm that generates a permutation of a given cycle type, then conjugate B A B' is also an algorithm that will generate another permutation of the *same* cycle type. This means that conjugating an algorithm will *not* change the cycle type of the associated permutation, or more explicitely if A generates a 7-cycle of corner-centers, then B A B' will also generate a 7-cycle of corner-centers. This is a very simple means of building a set of algorithms of a given cycle type from an already known 'seed' algorithm. The operation of conjugating an algorithm is also known as 'setup'. In most practical cases, setup moves are 1-, 2- or 3-move long. By using a set of different seed algorithms, it is possible to lower the number of needed setup moves from 3 to 2 or even 1.

Coset:
If A is an algorithm that generates a permutation of a given cycle type, then B A is also an algorithm that will generate another permutation, but which is generally of a *different* cycle type, although permutations of a same cycle type *may* be generated in *some* cases. This means that by inserting moves at the left, algorithms of a same cycle type *may* eventually be generated. Left added algorithms are usually 1-, 2- or 3-move long.

Symmetry:
If A is an algorithm that generates a permutation of a given cycle type, then transformed algorithm B(A), obtained by applying (anti)symmetry transformations on the set of letters <F, R, U, L, D, B> is also an algorithm that will generate another permutation of the *same* cycle type. This means that transforming an algorithm using all 48 possible cube (anti)symmetries will *not* change the cycle type of the associated permutations, or more explicitely if A generates a 5-cycle of edge-centers, then 48 more algorithms may be obtained by transforming letters.

Inversion:
If A is an algorithm that generates a permutation of a given cycle type, then inverted algorithm A' is also an algorithm that will generate another permutation of the *same* cycle type. This means that for each already known algorithm, one more can be added simply by inversion.

Cycle Shifting:
If A is an algorithm of length n that generates a permutation of a given cycle type, then the n algorithms obtained by cycle shifting A will also generate permutations of the *same* cycle type. Alg NR' U N3R U' NR U N3R' U' and the shifted version U' NR' U N3R U' NR U N3R' will *both* give a 3-cycle of edge-centers, for example.

Using a combination of all these techniques, ie. conjugation + left add + symmetry + inversion + cycle shifting, it is usually possible to generate many N-cycles of a given type from just a *limited* number of seed algorithms. The problem now is how to find *short* and *irreducible* seeds. This is where algorithm templates may come into play…

Semi-Commutators

Overview

It has already been demonstrated that all permutations of even parity can be written as commutators. But a more general way of describing a permutation of even parity would be as follows :

A (B C B')

where A, B and C are *multiple* move sequences, (B C B') is a conjugator and C is a function of A.

If C = A', then this expression gives the well-known commutator A B A' B' = [A, B]. We can go further by stating that C can also be a function of A, provided that the parity of the permutation is kept even. By decomposing A into a sequence of n basic moves:

A = A1 A2 A3 … An

sequence C can then be written as:

C = C1 C2 C3 … Cn

There are now 2 possible cases that will keep parity:

C = An' … A3' A2' A1'

C = A1' A2' A3' … An'

In the first case, we have the well-known commutator:

(A1 A2 A3 … An) . B . (An' … A3' A2' A1') . B' = [A, B]

whereas in the second case, it is a *semi*-commutator:

(A1 A2 A3 … An) . B . (A1' A2' A3' … An') . B' =]A, B]

There are just 4 ways of writing a semi-commutator:

(A1 A2) . (B1 B2) . (A1' A2') . (B2' B1') =]A1 A2, B1 B2]

(A1 A2) . (B1 B2) . (A1' A2') . (B1' B2') =]A1 A2, B1 B2[

(A1 A2) . (B1 B2) . (A2' A1') . (B1' B2') = [A1 A2, B1 B2[

(A1 A2) . (B1 B2) . (A2' A1') . (B2' B1') = [A1 A2, B1 B2]

We can see that semi-commutator #4 is actually a commutator, so that a semi-commutator may simply be seen as an attempt to generalize the commutator concept.

Example

Semi-commutators may be useful for finding algorithms. Using for example a template such as]X Y, Z P Q P'] to search for 15-cycles of corner-centers, will give the algorithm below, which is clearly not a commutator:

]NR' NF', NL L' ND L] = NR' NF' . NL L' ND L . NR NF . L' ND' L NL'

But if it is written as a commutator, then we have a 9-cycle of corner-centers:

[NR' NF', NL L' ND L] = NR' NF' . NL L' ND L . NF NR . L' ND' L NL'

Block Decomposition

Another way of understanding semi-commutators is from block decomposition of a sequence of moves. Consider for example a first block of n moves followed by a second block of the same length: there must be the same number of moves and inverted moves in the complete sequence of 2 blocks, in order to keep the permutation parity even. There are many ways of re-arranging moves and inverted moves in the sequence. In a simple arrangement, all non-inverted moves are placed in the first block and all inverted moves in the second block:

First block: A1 A2 A3 … An
Second block: same moves, but inverted, sequenced and partitioned differently

By using semi-commutators, there are just 4 different ways of re-arranging inverted moves in the second block.

This is shown in the table below, where a sequence of length 14 has been chosen and possible semi-commutator structures are listed.

Block partitioning – Semi-Commutators – 14-Move Sequence Example			
Index	First Block of Moves	Second Block of Inverted Moves	Semi-Commutators/Commutators
1	(X Y Z P Q V A)	(X' Y' Z' P' Q' V' A')	[X, Y Z P Q V A[
2	(X) (Y Z P Q V A)	(X') (Y' Z' P' Q' V' A')	[X, Y Z P Q V A[
3	(X) (Y Z P Q V A)	(X') (A' V' Q' P' Z' Y')	[X, Y Z P Q V A]
4	(X Y) (Z P Q V A)	(X' Y') (Z' P' Q' V' A')	]X Y, Z P Q V A[
5	(X Y) (Z P Q V A)	(X' Y') (A' V' Q' P' Z')	]X Y, Z P Q V A]
6	(X Y) (Z P Q V A)	(Y' X') (Z' P' Q' V' A')	[X Y, Z P Q V A[
7	(X Y) (Z P Q V A)	(Y' X') (A' V' Q' P' Z')	[X Y, Z P Q V A]
8	(X Y Z) (P Q V A)	(X' Y' Z') (P' Q' V' A')	]X Y Z, P Q V A[
9	(X Y Z) (P Q V A)	(X' Y' Z') (A' V' Q' P')	]X Y Z, P Q V A]
10	(X Y Z) (P Q V A)	(Z' Y' X') (P' Q' V' A')	[X Y Z, P Q V A[
11	(X Y Z) (P Q V A)	(Z' Y' X') (A' V' Q' P')	[X Y Z, P Q V A]

Semi-commutator #6 is then written as:

[X Y, Z P Q V A[= X Y . Z P Q V A . Y' X' . Z' P' Q' V' A'

Whereas commutator #7 would read:

[X Y, Z P Q V A] = X Y . Z P Q V A . Y' X' . A' V' Q' P' Z'

Usefulness

Semi-commutators are mainly used in searching for new algorithms and building arrays of seeds. For permutations involving more than 3 pieces, templates built on semi-commutators will usually give additional algorithms that will complement algorithms already found using commutators only.

Algorithm Templates

Algorithm Templates – Corners Two 2-Cycles	
Template	Moves
[X, Y Z P Q V]	12
[X, Y Z P Q V[	12
(X Y Z P Z' Y')2 X2	13
(X Y Z P Y' Z')2 X2	13
(X Y X') Z (X Y' X') (P Q P') Z' (P Q' P')	14
Set of Moves – 18 Moves in Set	
<R, R', R2, U, U', U2, F, F', F2, L, L', L2, D, D', D2, B, B', B2>	
Number of Permuted Pieces	Permutation Order
4	2

Algorithm Templates – Midges Two 2-Cycles	
Template	Moves
[X, Y Z P Q]	10
[X, Y Z P Q[	10
[X, Y Z P Z' Y']	12
[X, Y Z P Y' Z']	12
[X, Y Z P Z' Y'[	12
[X, Y Z P Y' Z'[	12
Set of Moves – 36 Moves in Set	
<R, R', R2, U, U', U2, F, F', F2, L, L', L2, D, D', D2, B, B', B2,	
MR, MR', MR2, MU, MU', MU2, MF, MF', MF2, ML, ML', ML2, MD, MD', MD2, MB, MB', MB2>	
Number of Permuted Pieces	Permutation Order
4	2

Algorithm Templates – Edges Two 2-Cycles	
Template	Moves
[X, Y Z P Z' Y']	12
[X, Y Z P Y' Z']	12
[X, Y Z P Z' Y'[	12
[X, Y Z P Y' Z'[	12
(X Y Z P Z' Y')2 X2	13
(X Y Z P Y' Z')2 X2	13
(X Y X') Z (X Y' X') (P Q P') Z' (P Q' P')	14
Set of Moves – 36 Moves in Set	
<R, R', R2, U, U', U2, F, F', F2, L, L', L2, D, D', D2, B, B', B2,	
NR, NR', NR2, NU, NU', NU2, NF, NF', NF2, NL, NL', NL2, ND, ND', ND2, NB, NB', NB2>	
Number of Permuted Pieces	Permutation Order
4	2

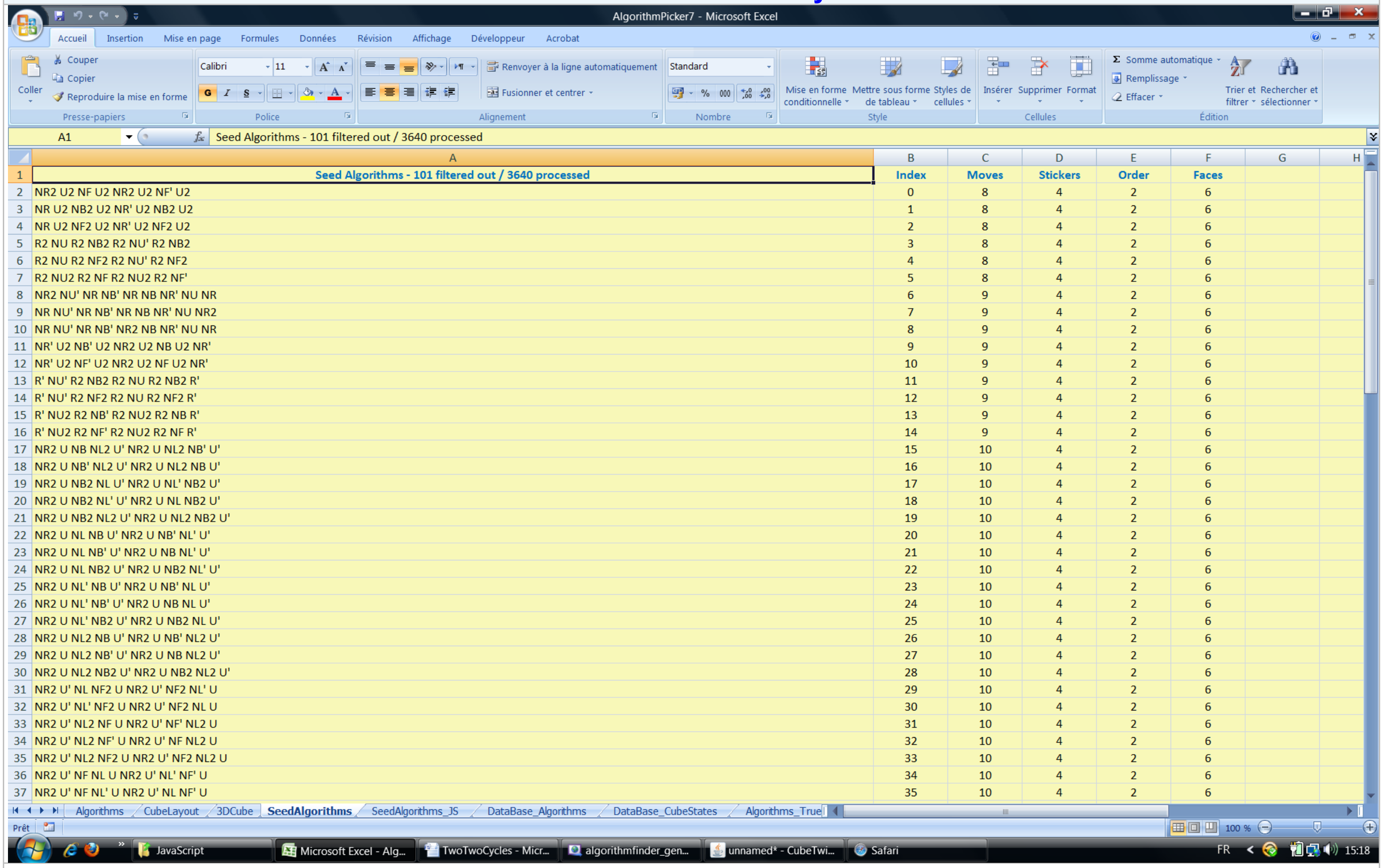
Algorithm Templates – Corner-Centers Two 2-Cycles	
Template	Moves
[X, Y Z P]	8
[X, Y Z P[	8
[X, Y Z P Q]	10
[X, Y Z P Q[	10
Set of Moves – 36 Moves in Set	
<R, R', R2, U, U', U2, F, F', F2, L, L', L2, D, D', D2, B, B', B2,	
NR, NR', NR2, NU, NU', NU2, NF, NF', NF2, NL, NL', NL2, ND, ND', ND2, NB, NB', NB2>	
Number of Permuted Pieces	Permutation Order
4	2

Algorithm Templates – Midge-Centers Two 2-Cycles	
Template	Moves
[X, Y]	4
[X, Y Z]	6
[X, Y Z[	6
Set of Moves – 48 Moves in Set	
<R, R', R2, U, U', U2, F, F', F2, L, L', L2, D, D', D2, B, B', B2,	
MR, MR', MR2, MU, MU', MU2, MF, MF', MF2, ML, ML', ML2, MD, MD', MD2, MB, MB', MB2,	
NR, NR', NR2, NU, NU', NU2, NF, NF', NF2, NL, NL', NL2, ND, ND', ND2, NB, NB', NB2>	
Number of Permuted Pieces	Permutation Order
4	2

Algorithm Templates – Edge-Centers Two 2-Cycles	
Template	Moves
[X, Y Z P]	8
[X, Y Z P[	8
[X, Y Z P Q]	10
[X, Y Z P Q[	10
Set of Moves – 48 Moves in Set	
<R, R', R2, U, U', U2, F, F', F2, L, L', L2, D, D', D2, B, B', B2,	
NR, NR', NR2, NU, NU', NU2, NF, NF', NF2, NL, NL', NL2, ND, ND', ND2, NB, NB', NB2,	
N3R, N3R', N3R2, N3U, N3U', N3U2, N3F, N3F', N3F2, N3L, N3L', N3L2, N3D, N3D', N3D2, N3B, N3B', N3B2>	
Number of Permuted Pieces	Permutation Order
4	2

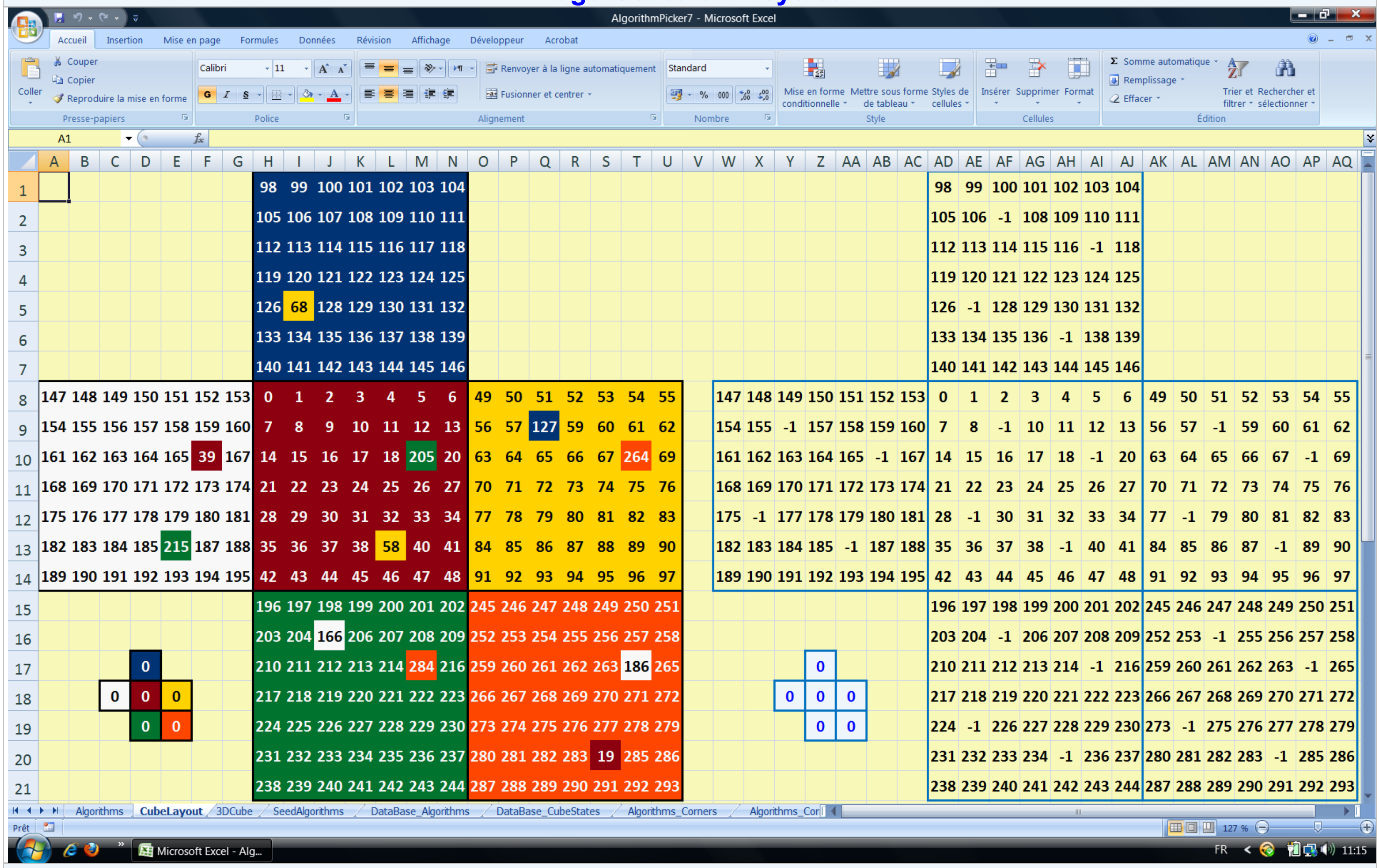
Search for irreducible seed algorithms using Algorithm Finder – 7x7x7 Cube

Corner-Centers Two 2-cycles



	A	B	C	D	E	F
1	Seed Algorithms - 101 filtered out / 3640 processed	Index	Moves	Stickers	Order	Faces
2	NR2 U2 NF U2 NR2 U2 NF' U2	0	8	4	2	6
3	NR U2 NB2 U2 NR' U2 NB2 U2	1	8	4	2	6
4	NR U2 NF2 U2 NR' U2 NF2 U2	2	8	4	2	6
5	R2 NU R2 NB2 R2 NU' R2 NB2	3	8	4	2	6
6	R2 NU R2 NF2 R2 NU' R2 NF2	4	8	4	2	6
7	R2 NU2 R2 NF R2 NU2 R2 NF'	5	8	4	2	6
8	NR2 NU' NR NB' NR NB NR' NU NR	6	9	4	2	6
9	NR NU' NR NB' NR NB NR' NU NR2	7	9	4	2	6
10	NR NU' NR NB' NR2 NB NR' NU NR	8	9	4	2	6
11	NR' U2 NB' U2 NR2 U2 NB U2 NR'	9	9	4	2	6
12	NR' U2 NF' U2 NR2 U2 NF U2 NR'	10	9	4	2	6
13	R' NU' R2 NB2 R2 NU R2 NB2 R'	11	9	4	2	6
14	R' NU' R2 NF2 R2 NU R2 NF2 R'	12	9	4	2	6
15	R' NU2 R2 NB' R2 NU2 R2 NB R'	13	9	4	2	6
16	R' NU2 R2 NF' R2 NU2 R2 NF R'	14	9	4	2	6
17	NR2 U NB NL2 U' NR2 U NL2 NB' U'	15	10	4	2	6
18	NR2 U NB' NL2 U' NR2 U NL2 NB U'	16	10	4	2	6
19	NR2 U NB2 NL U' NR2 U NL' NB2 U'	17	10	4	2	6
20	NR2 U NB2 NL' U' NR2 U NL NB2 U'	18	10	4	2	6
21	NR2 U NB2 NL2 U' NR2 U NL2 NB2 U'	19	10	4	2	6
22	NR2 U NL NB U' NR2 U NB' NL' U'	20	10	4	2	6
23	NR2 U NL NB' U' NR2 U NB NL' U'	21	10	4	2	6
24	NR2 U NL NB2 U' NR2 U NB2 NL' U'	22	10	4	2	6
25	NR2 U NL' NB U' NR2 U NB' NL U'	23	10	4	2	6
26	NR2 U NL' NB' U' NR2 U NB NL U'	24	10	4	2	6
27	NR2 U NL' NB2 U' NR2 U NB2 NL U'	25	10	4	2	6
28	NR2 U NL2 NB U' NR2 U NB' NL2 U'	26	10	4	2	6
29	NR2 U NL2 NB' U' NR2 U NB NL2 U'	27	10	4	2	6
30	NR2 U NL2 NB2 U' NR2 U NB2 NL2 U'	28	10	4	2	6
31	NR2 U' NL NF2 U NR2 U' NF2 NL' U	29	10	4	2	6
32	NR2 U' NL' NF2 U NR2 U' NF2 NL U	30	10	4	2	6
33	NR2 U' NL2 NF U NR2 U' NF' NL2 U	31	10	4	2	6
34	NR2 U' NL2 NF' U NR2 U' NF NL2 U	32	10	4	2	6
35	NR2 U' NL2 NF2 U NR2 U' NF2 NL2 U	33	10	4	2	6
36	NR2 U' NF NL U NR2 U' NL' NF' U	34	10	4	2	6
37	NR2 U' NF NL' U NR2 U' NL NF' U	35	10	4	2	6

Edge-Centers 11-cycles



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ
1								98	99	100	101	102	103	104																98	99	100	101	102	103	104							
2								105	106	107	108	109	110	111																105	106	-1	108	109	110	111							
3								112	113	114	115	116	117	118																112	113	114	115	116	-1	118							
4								119	120	121	122	123	124	125																119	120	121	122	123	124	125							
5								126	68	128	129	130	131	132																126	-1	128	129	130	131	132							
6								133	134	135	136	137	138	139																133	134	135	136	-1	138	139							
7								140	141	142	143	144	145	146																140	141	142	143	144	145	146							
8	147	148	149	150	151	152	153	0	1	2	3	4	5	6	49	50	51	52	53	54	55		147	148	149	150	151	152	153	0	1	2	3	4	5	6	49	50	51	52	53	54	55
9	154	155	156	157	158	159	160	7	8	9	10	11	12	13	56	57	127	59	60	61	62		154	155	-1	157	158	159	160	7	8	-1	10	11	12	13	56	57	-1	59	60	61	62
10	161	162	163	164	165	39	167	14	15	16	17	18	205	20	63	64	65	66	67	264	69		161	162	163	164	165	-1	167	14	15	16	17	18	-1	20	63	64	65	66	67	-1	69
11	168	169	170	171	172	173	174	21	22	23	24	25	26	27	70	71	72	73	74	75	76		168	169	170	171	172	173	174	21	22	23	24	25	26	27	70	71	72	73	74	75	76
12	175	176	177	178	179	180	181	28	29	30	31	32	33	34	77	78	79	80	81	82	83		175	-1	177	178	179	180	181	28	-1	30	31	32	33	34	77	-1	79	80	81	82	83
13	182	183	184	185	215	187	188	35	36	37	38	58	40	41	84	85	86	87	88	89	90		182	183	184	185	-1	187	188	35	36	37	38	-1	40	41	84	85	86	87	-1	89	90
14	189	190	191	192	193	194	195	42	43	44	45	46	47	48	91	92	93	94	95	96	97		189	190	191	192	193	194	195	42	43	44	45	46	47	48	91	92	93	94	95	96	97
15								196	197	198	199	200	201	202	245	246	247	248	249	250	251									196	197	198	199	200	201	202	245	246	247	248	249	250	251
16								203	204	166	206	207	208	209	252	253	254	255	256	257	258									203	204	-1	206	207	208	209	252	253	-1	255	256	257	258
17				0				210	211	212	213	214	284	216	259	260	261	262	263	186	265					0				210	211	212	213	214	-1	216	259	260	261	262	263	-1	265
18			0	0	0			217	218	219	220	221	222	223	266	267	268	269	270	271	272				0	0	0			217	218	219	220	221	222	223	266	267	268	269	270	271	272
19				0	0			224	225	226	227	228	229	230	273	274	275	276	277	278	279					0	0			224	-1	226	227	228	229	230	273	-1	275	276	277	278	279
20								231	232	233	234	235	236	237	280	281	282	283	19	285	286									231	232	233	234	-1	236	237	280	281	282	283	-1	285	286
21								238	239	240	241	242	243	244	287	288	289	290	291	292	293									238	239	240	241	242	243	244	287	288	289	290	291	292	293

All stickers of the first orbit of edge-centers have been set to -1 prior searching for 11-cycles.