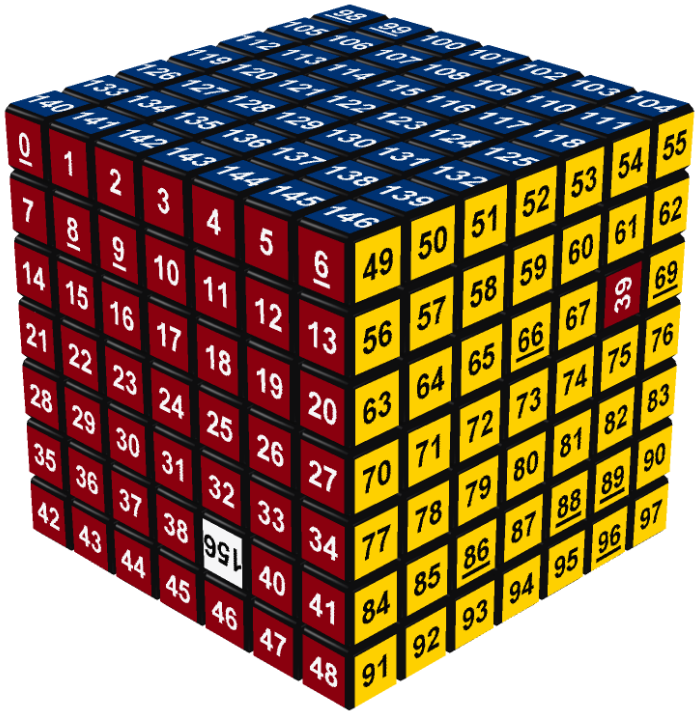
Seed Algorithms

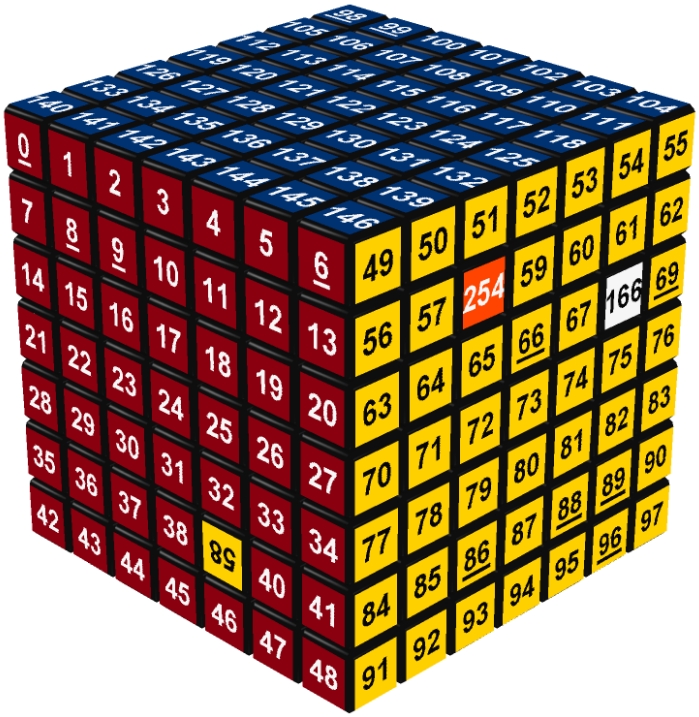
Introduction

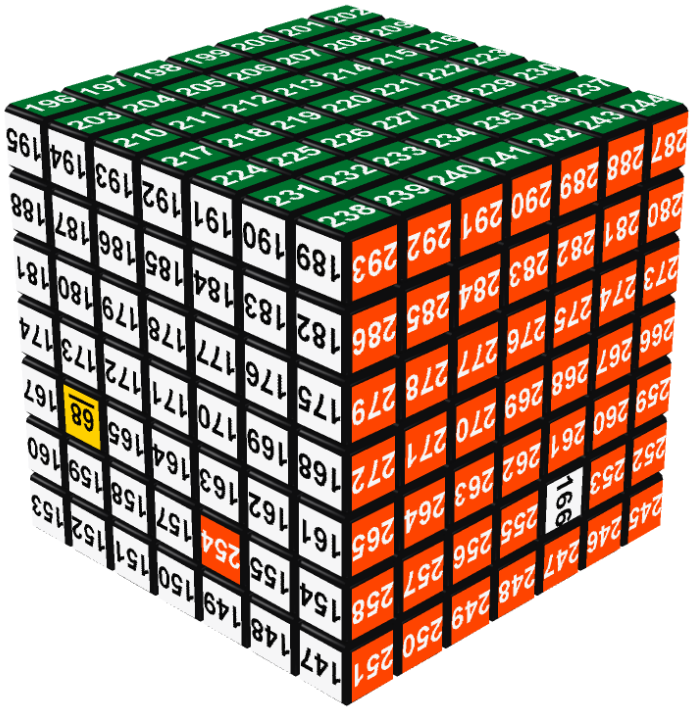
N-cycles are particular permutations of cube pieces where N pieces are permuted in a *single* cycle. Among these are even parity permutations, for which an *odd* number of pieces are permuted. It may be of interest to find short algorithms for cycling up to 7 corners, 11 midges, 23 edges and 23 centers, without messing up other pieces.

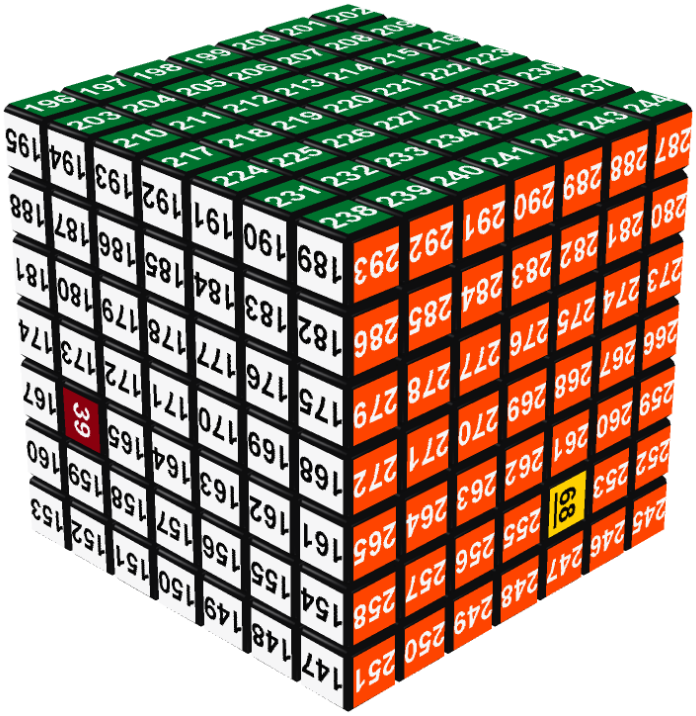
In this context, a *seed* algorithm can be defined as a short algorithm that could generate a set of algorithms, by rotation/reflection, inversion, cyclic-shift, adding 1, 2 or 3 moves to the left or 1, 2 or 3 setup moves. Seed algorithms are *irreducible*, meaning that they can't be obtained from each other by inversion, cyclic-shift or symmetry considerations (see: http://www.mementoslangues.fr/CubeDesign/CubeTheory/CubeSymmetry.pdf).

Irreducible Seed Algorithms – Edge-Centers – 5-Cycle









[N3R2 N3U2, L' NU' L NU]

[N3R2 N3U2, R' NU R NU']

Although quite similar, these 2 algorithms can't be obtained from each other by cube symmetry considerations.

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 usually 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 simply 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-Shift:**
  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. This means for example that 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.

Using a combination of all these techniques, ie. conjugation + left add + symmetry + inversion + cycle-shift, 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 can 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 – Corner-Centers			
Template	Inverted Template	N-cycle	Moves
[X, Y Z P]	[X Y Z, P]	3-, 5-, 9-cycle	8
[X, Y Z P Q]	[X Y Z P, Q]	7-, 11-, 13-cycle	10
[X Y, Z P Q]	[X Y Z, P Q]	17-cycle	10
[X Y, Z P Q V]	[X Y Z P, Q V]	23-cycle	12
]X Y, Z P Q V]	[X Y Z P, Q V[	15-cycle	12
[X Y, Z P Q V A]	[X Y Z P Q, V A]	19-, 21-cycle	14

Algorithm Templates – Edge-Centers			
Template	Inverted Template	N-cycle	Moves
[X, Y Z P]	[X Y Z, P]	3-cycle	8
[X Y, Z P Q V]	[X Y Z P, Q V]	5-cycle	12
[X Y, Z P Q V A]	[X Y Z P Q, V A]	7-cycle	14
[X Y, Z P Q V A G]	[X Y Z P Q V, A G]	9-cycle	16

Note:
]X Y, Z P Q V] = X Y . Z P Q V . X' Y' . V' Q' P' Z' is *not* a commutator
[X Y, Z P Q V] = X Y . Z P Q V . Y' X' . V' Q' P' Z' *is* a commutator

[X Y Z P, Q V[= X Y Z P . Q V . P' Z' Y' X' . Q' V' is *not* a commutator
[X Y, Z P Q V] = X Y Z P . Q V . P' Z' Y' X' . V' Q' *is* a commutator

Seed Algorithms

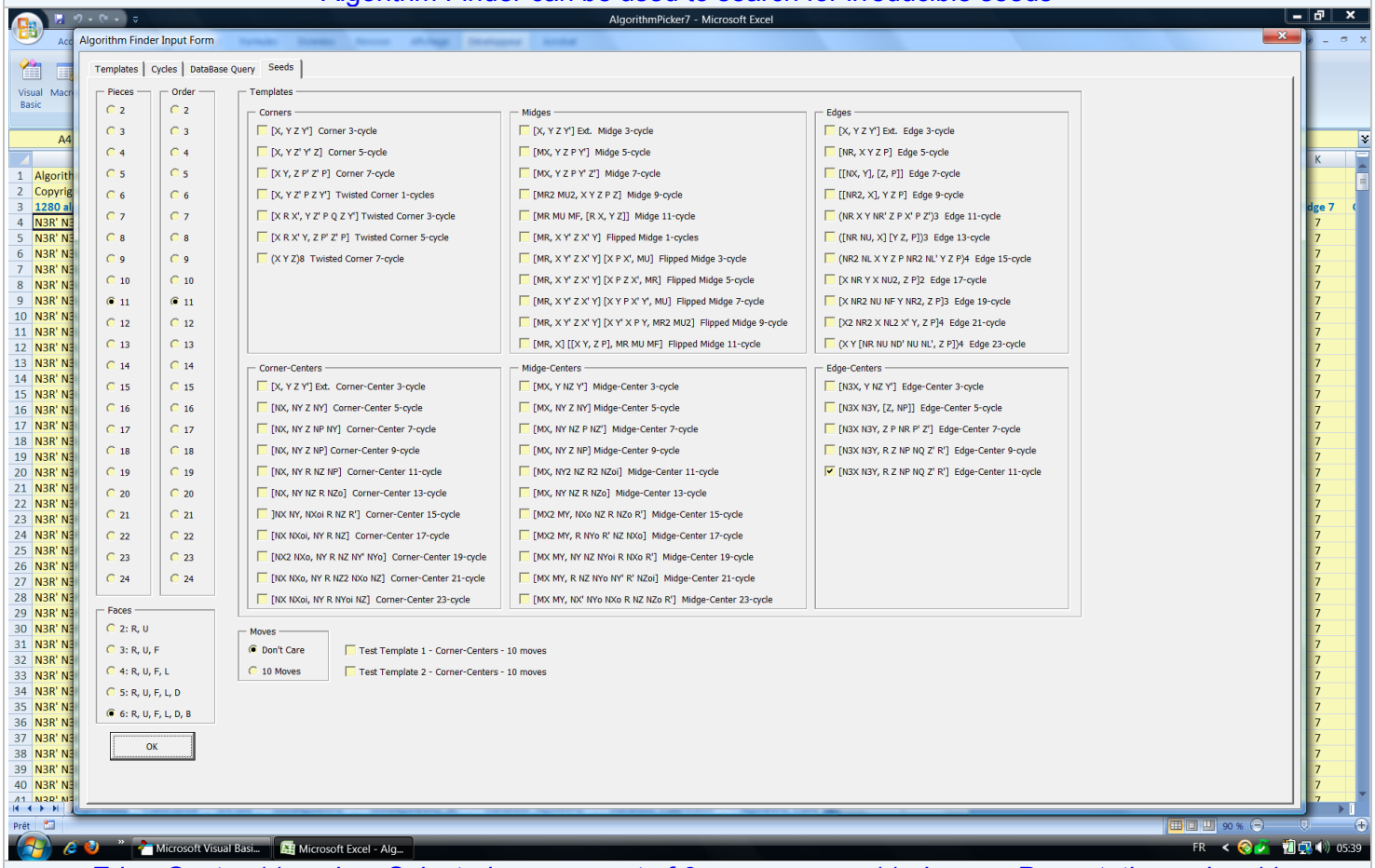
Seed Algorithms – Corner-Centers			
Commutator General Form: [N2 moves, Face + N2 moves]			
Template / Inverted Template	Seed Algorithms – Examples	N-cycle	Moves
[X, Y Z P] / [X Y Z, P]	[NR, F NL F']	3-cycle	8
[X, Y Z P] / [X Y Z, P]	[NU, NL' U' NL']	5-cycle	8
[X, Y Z P Q] / [X Y Z P, Q]	[NR', NU' L2 NF' NU']	7-cycle	10
[X, Y Z P] / [X Y Z, P]	[NB, NR B2 NU]	9-cycle	8
[X, Y Z P Q] / [X Y Z P, Q]	[NF', NR' F NL' NU']	11-cycle	10
[X, Y Z P Q] / [X Y Z P, Q]	[NL', ND' NF' L' NB']	13-cycle	10
]X Y, Z P Q V] / [X Y Z P, Q V[	]NR' NF', NL L' ND L]	15-cycle	12
[X Y, Z P Q] / [X Y Z, P Q]	[NB NF', ND F' NR]	17-cycle	10
[X Y, Z P Q V A] / [X Y Z P Q, V A]	[NF2 NB', ND' F' NL ND NU']	19-cycle	14
[X Y, Z P Q V A] / [X Y Z P Q, V A]	[NB NF, NU F NL2 NF NL]	21-cycle	14
[X Y, Z P Q V] / [X Y Z P, Q V]	[NB NF', NR B' NL' ND]	23-cycle	12
2-Cycle + 2-Cycle			
[X, Y Z P] / [X Y Z, P]	[NR, U2 NF2 U2]	2c + 2c	8
[X, Y Z P[/]X Y Z, P]	[NU, F2 NR2 F2[	2c + 2c	8
]X Y, Z P[/]X Y, Z P[	]R2 NU, R2 NF2[	2c + 2c	8
4-Cycle + 2-Cycle			
[X, Y Z P Q] / [X Y Z P, Q]	[NR2, U' NL2 U NF2]	4c + 2c	10
[X Y, Z P Q] / [X Y Z, P Q]	[NR U', NL NU2 NL']	4c + 2c	10

Seed Algorithms – Edge-Centers			
Commutator General Form: [N3 moves, Face + N2 moves]			
Template / Inverted Template	Seed Algorithms – Examples	N-cycle	Moves
[X, Y Z P] / [X Y Z, P]	[N3R', U NR U']	3-cycle	8
[X Y, Z P Q V] / [X Y Z P, Q V]	[N3R N3U, F' NR F NR']	5-cycle	12
[X Y, Z P Q V A] / [X Y Z P Q, V A]	[N3R2 N3F2, F' U' NR' U F]	7-cycle	14
[X Y, Z P Q V A G] / [X Y Z P Q V, A G]	[N3R N3U, R' F2 NU NL2 F2 R]	9-cycle	16
2-Cycle + 2-Cycle			
[X, Y Z P] / [X Y Z, P]	[N3R, F2 NU2 F2]	2c + 2c	8
[X, Y Z P[/]X Y Z, P]	[N3F2, U2 NR' U2[	2c + 2c	8
4-Cycle + 2-Cycle			
[X Y, Z P Q V A] / [X Y Z P Q, V A]	[N3R N3U, R' F' NU F R]	4c + 2c	14

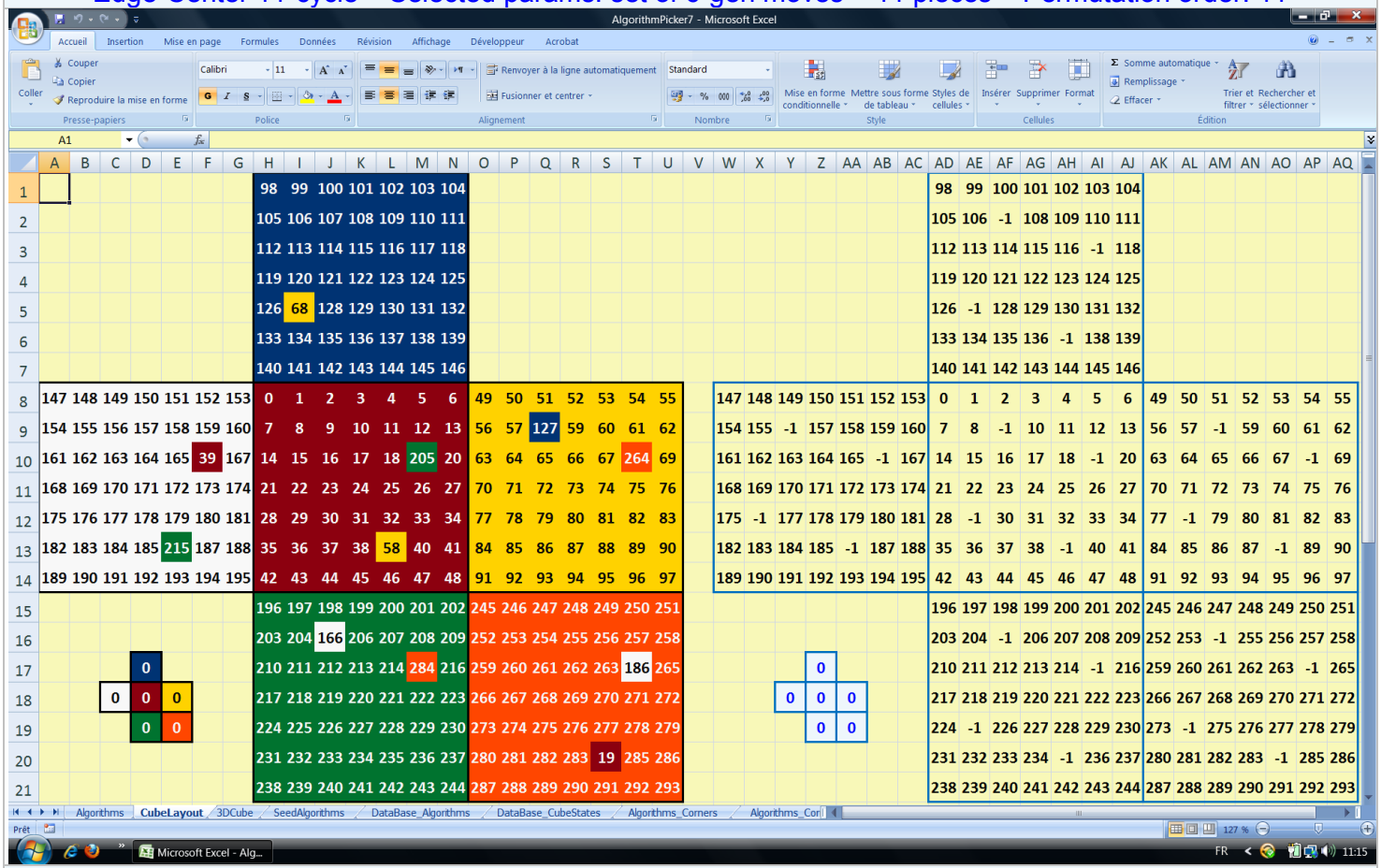
Algorithm Finder

Search for Irreducible Seed Algorithms – 1

Algorithm Finder can be used to search for irreducible seeds



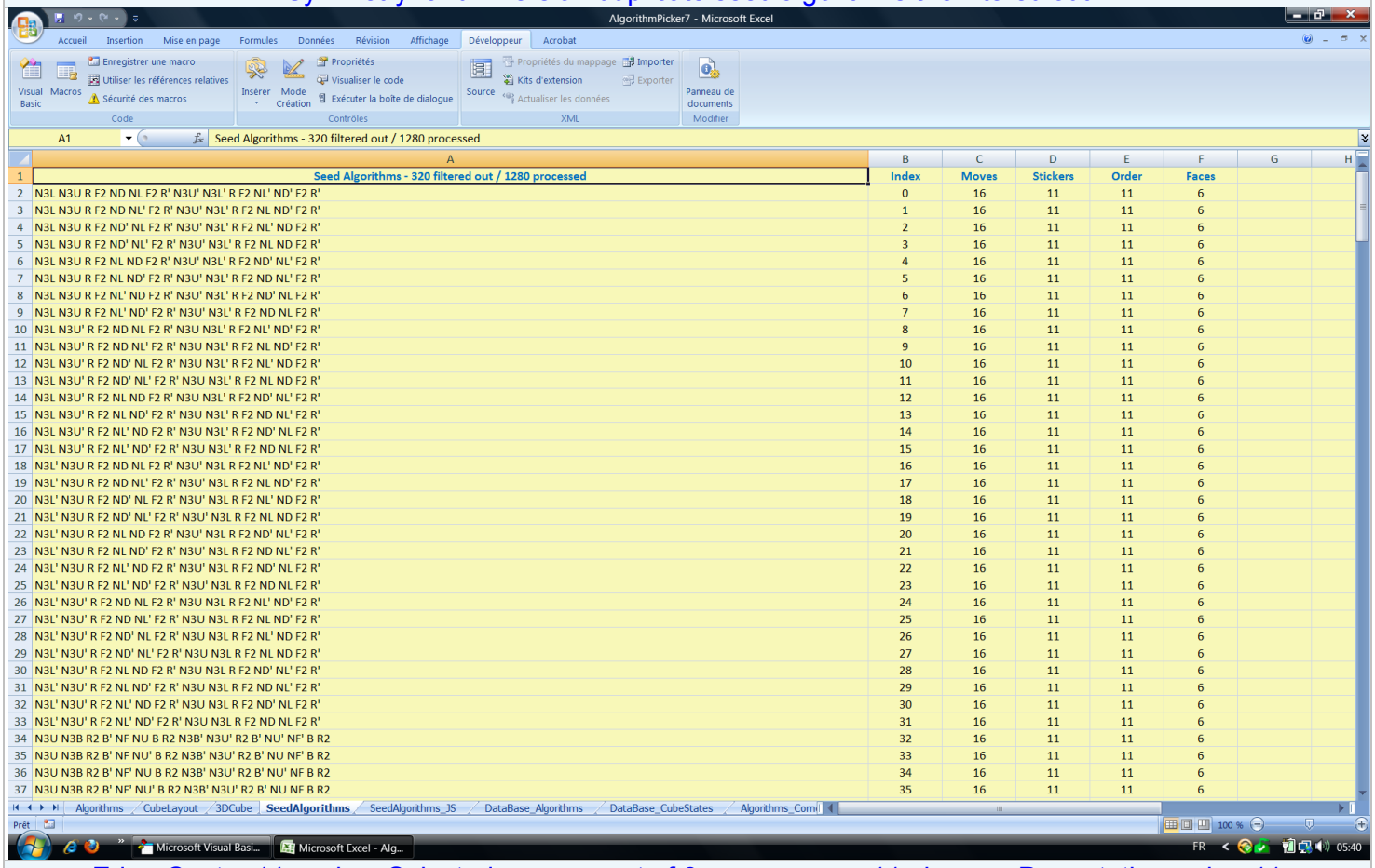
Edge-Center 11-cycle – Selected params: set of 6-gen moves – 11 pieces – Permutation order: 11



All stickers of the first orbit of edge-centers have been set to -1. Algorithms are then filtered out both by permutation order and by number of twisted/flipped/moved pieces.

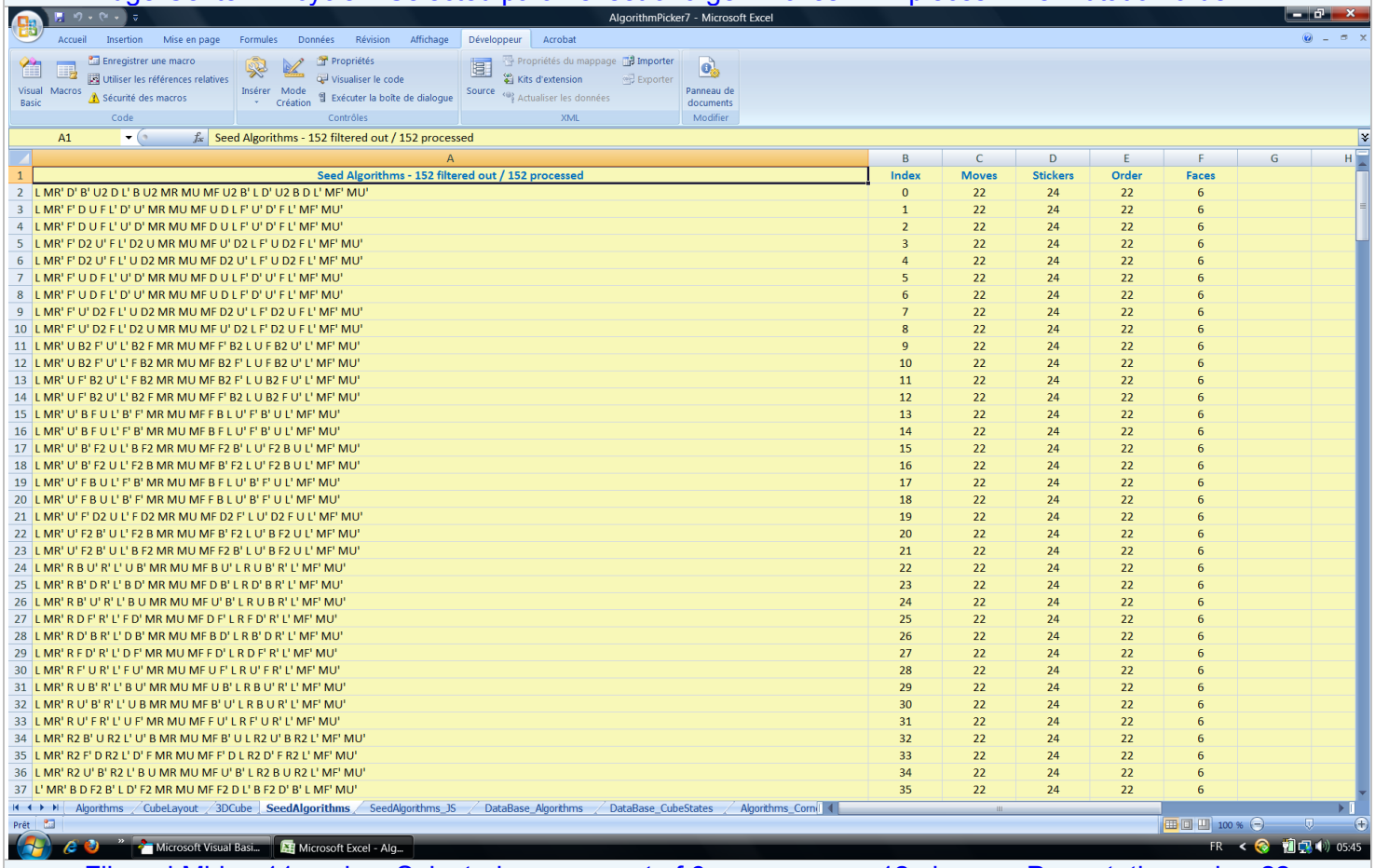
Search for Irreducible Seed Algorithms – 2

Symmetry- and inversion-duplicate seed algorithms are filtered out



Seed Algorithms - 320 filtered out / 1280 processed	Index	Moves	Stickers	Order	Faces
N3L N3U R F2 ND NL F2 R' N3U' N3L' R F2 NL' ND' F2 R'	0	16	11	11	6
N3L N3U R F2 ND NL' F2 R' N3U' N3L' R F2 NL ND' F2 R'	1	16	11	11	6
N3L N3U R F2 ND' NL F2 R' N3U' N3L' R F2 NL' ND F2 R'	2	16	11	11	6
N3L N3U R F2 ND' NL' F2 R' N3U' N3L' R F2 NL ND F2 R'	3	16	11	11	6
N3L N3U R F2 NL ND F2 R' N3U' N3L' R F2 ND' NL' F2 R'	4	16	11	11	6
N3L N3U R F2 NL ND' F2 R' N3U' N3L' R F2 ND NL' F2 R'	5	16	11	11	6
N3L N3U R F2 NL' ND F2 R' N3U' N3L' R F2 ND' NL F2 R'	6	16	11	11	6
N3L N3U R F2 NL' ND' F2 R' N3U' N3L' R F2 ND NL F2 R'	7	16	11	11	6
N3L N3U' R F2 ND NL F2 R' N3U N3L' R F2 NL' ND' F2 R'	8	16	11	11	6
N3L N3U' R F2 ND NL' F2 R' N3U N3L' R F2 NL ND' F2 R'	9	16	11	11	6
N3L N3U' R F2 ND' NL F2 R' N3U N3L' R F2 NL' ND F2 R'	10	16	11	11	6
N3L N3U' R F2 ND' NL' F2 R' N3U N3L' R F2 NL ND F2 R'	11	16	11	11	6
N3L N3U' R F2 NL ND F2 R' N3U N3L' R F2 ND' NL' F2 R'	12	16	11	11	6
N3L N3U' R F2 NL ND' F2 R' N3U N3L' R F2 ND NL' F2 R'	13	16	11	11	6
N3L N3U' R F2 NL' ND F2 R' N3U N3L' R F2 ND' NL F2 R'	14	16	11	11	6
N3L N3U' R F2 NL' ND' F2 R' N3U N3L' R F2 ND NL F2 R'	15	16	11	11	6
N3L' N3U R F2 ND NL F2 R' N3U' N3L R F2 NL' ND' F2 R'	16	16	11	11	6
N3L' N3U R F2 ND NL' F2 R' N3U' N3L R F2 NL ND' F2 R'	17	16	11	11	6
N3L' N3U R F2 ND' NL F2 R' N3U' N3L R F2 NL' ND F2 R'	18	16	11	11	6
N3L' N3U R F2 ND' NL' F2 R' N3U' N3L R F2 NL ND F2 R'	19	16	11	11	6
N3L' N3U R F2 NL ND F2 R' N3U' N3L R F2 ND' NL' F2 R'	20	16	11	11	6
N3L' N3U R F2 NL ND' F2 R' N3U' N3L R F2 ND NL' F2 R'	21	16	11	11	6
N3L' N3U R F2 NL' ND F2 R' N3U' N3L R F2 ND' NL F2 R'	22	16	11	11	6
N3L' N3U R F2 NL' ND' F2 R' N3U' N3L R F2 ND NL F2 R'	23	16	11	11	6
N3L' N3U R F2 ND NL F2 R' N3U' N3L R F2 NL' ND F2 R'	24	16	11	11	6
N3L' N3U' R F2 ND NL' F2 R' N3U N3L R F2 NL ND' F2 R'	25	16	11	11	6
N3L' N3U' R F2 ND' NL F2 R' N3U N3L R F2 NL' ND F2 R'	26	16	11	11	6
N3L' N3U' R F2 ND' NL' F2 R' N3U N3L R F2 NL ND F2 R'	27	16	11	11	6
N3L' N3U' R F2 NL ND F2 R' N3U N3L R F2 ND' NL' F2 R'	28	16	11	11	6
N3L' N3U' R F2 NL ND' F2 R' N3U N3L R F2 ND NL' F2 R'	29	16	11	11	6
N3L' N3U' R F2 NL' ND F2 R' N3U N3L R F2 ND' NL F2 R'	30	16	11	11	6
N3L' N3U' R F2 NL' ND' F2 R' N3U N3L R F2 ND NL F2 R'	31	16	11	11	6
N3U N3B R2 B' NF NU B R2 N3B' N3U' R2 B' NU' NF' B R2	32	16	11	11	6
N3U N3B R2 B' NF NU' B R2 N3B' N3U' R2 B' NU NF' B R2	33	16	11	11	6
N3U N3B R2 B' NF' NU B R2 N3B' N3U' R2 B' NU' NF B R2	34	16	11	11	6
N3U N3B R2 B' NF' NU' B R2 N3B' N3U' R2 B' NU NF B R2	35	16	11	11	6

Edge-Center 11-cycle – Selected params: set of 6-gen moves – 11 pieces – Permutation order: 11



Seed Algorithms - 152 filtered out / 152 processed	Index	Moves	Stickers	Order	Faces
L MR' D' B' U2 D L' B U2 MR MU MF U2 B' L D' U2 B D L' MF' MU'	0	22	24	22	6
L MR' F' D U F L' D' U' MR MU MF U D L F' U' D' F L' MF' MU'	1	22	24	22	6
L MR' F' D U F L' U' D' MR MU MF D U L F' U' D' F L' MF' MU'	2	22	24	22	6
L MR' F' D2 U' F L' D2 U MR MU MF U' D2 L F' U D2 F L' MF' MU'	3	22	24	22	6
L MR' F' D2 U' F L' U D2 MR MU MF D2 U' L F' U D2 F L' MF' MU'	4	22	24	22	6
L MR' F' U D F L' U' D' MR MU MF D U L F' D' U' F L' MF' MU'	5	22	24	22	6
L MR' F' U D F L' D' U' MR MU MF U D L F' D' U' F L' MF' MU'	6	22	24	22	6
L MR' F' U' D2 F L' U D2 MR MU MF D2 U' L F' D2 U F L' MF' MU'	7	22	24	22	6
L MR' F' U' D2 F L' D2 U MR MU MF U' D2 L F' D2 U F L' MF' MU'	8	22	24	22	6
L MR' U B2 F' U' L' B2 F MR MU MF F' B2 L U F B2 U' L' MF' MU'	9	22	24	22	6
L MR' U B2 F' U' L' F B2 MR MU MF B2 F' L U F B2 U' L' MF' MU'	10	22	24	22	6
L MR' U F' B2 U' L' F B2 MR MU MF B2 F' L U B2 F U' L' MF' MU'	11	22	24	22	6
L MR' U F' B2 U' L' B2 F MR MU MF F' B2 L U B2 F U' L' MF' MU'	12	22	24	22	6
L MR' U' B F U L' B' F' MR MU MF F B L U' F' B' U L' MF' MU'	13	22	24	22	6
L MR' U' B F U L' F' B' MR MU MF B F L U' F' B' U L' MF' MU'	14	22	24	22	6
L MR' U' B' F2 U L' B F2 MR MU MF F2 B' L U' F2 B U L' MF' MU'	15	22	24	22	6
L MR' U' B' F2 U L' F2 B MR MU MF B' F2 L U' F2 B U L' MF' MU'	16	22	24	22	6
L MR' U' F B U L' F' B' MR MU MF B F L U' B' F' U L' MF' MU'	17	22	24	22	6
L MR' U' F B U L' B' F' MR MU MF F B L U' B' F' U L' MF' MU'	18	22	24	22	6
L MR' U' F' D2 U L' F D2 MR MU MF D2 F' L U' D2 F U L' MF' MU'	19	22	24	22	6
L MR' U' F2 B' U L' F2 B MR MU MF B' F2 L U' B F2 U L' MF' MU'	20	22	24	22	6
L MR' U' F2 B' U L' B F2 MR MU MF F2 B' L U' B F2 U L' MF' MU'	21	22	24	22	6
L MR' R B U' R' L' U B' MR MU MF B U' L R U B' R' L' MF' MU'	22	22	24	22	6
L MR' R B' D R' L' B D' MR MU MF D B' L R D' B R' L' MF' MU'	23	22	24	22	6
L MR' R B' U' R' L' B U MR MU MF U' B' L R U B R' L' MF' MU'	24	22	24	22	6
L MR' R D F' R' L' F D' MR MU MF D F' L R F D' R' L' MF' MU'	25	22	24	22	6
L MR' R D' B R' L' D B' MR MU MF B D' L R B' D R' L' MF' MU'	26	22	24	22	6
L MR' R F D' R' L' D F' MR MU MF F D' L R D F' R' L' MF' MU'	27	22	24	22	6
L MR' R F' U R' L' F U' MR MU MF U F' L R U' F R' L' MF' MU'	28	22	24	22	6
L MR' R U B' R' L' B U' MR MU MF U B' L R B U' R' L' MF' MU'	29	22	24	22	6
L MR' R U' B' R' L' U B MR MU MF B' U' L R B U R' L' MF' MU'	30	22	24	22	6
L MR' R U' F R' L' U F' MR MU MF F U' L R F' U R' L' MF' MU'	31	22	24	22	6
L MR' R2 B' U R2 L' U' B MR MU MF B' U L R2 U' B R2 L' MF' MU'	32	22	24	22	6
L MR' R2 F' D R2 L' D' F MR MU MF F' D L R2 D' F R2 L' MF' MU'	33	22	24	22	6
L MR' R2 U' B' R2 L' B U MR MU MF U' B' L R2 B U R2 L' MF' MU'	34	22	24	22	6
L' MR' B D F2 B' L D' F2 MR MU MF F2 D L' B F2 D' B' L MF' MU'	35	22	24	22	6

Flipped Midge 11-cycle – Selected params: set of 6-gen moves – 12 pieces – Permutation order: 22