Notes on the Kilominx

Tomas Rokicki
rokicki@gmail.com

February 16, 2026

Abstract

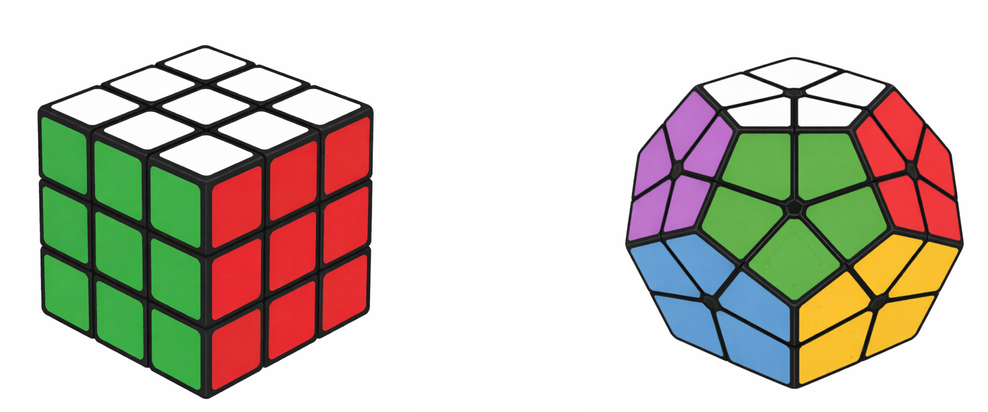
God's number for the Rubik's cube was determined in 2010, nearly 16 years ago. A state space of $4.3 * 10^{19}$ and the features of the group structure barely allowed its computation. Now, 16 years later, we begin to look at a puzzle with a slightly larger state space. This paper reviews current knowledge and lists challenges for the future.

1 Introduction

Following the progression of the Pocket Cube or 2x2x2, then the Rubik's Cube or 3x3x3, the logical next puzzle to investigate would seem to be the Rubik's Revenge (the 4x4x4). Unfortunately, the state space of the Revenge is so great (approximately $7.4 * 10^{45}$) that, with currently known techniques, we have little hope of solving even a single random position optimally, let alone determining the group diameter. So instead we focus first on an easier puzzle, the kilominx, with a state space of $2.4 * 10^{25}$, as a more reasonable challenge. Throughout we will relate the analysis here to that done for the cube. Most of the techniques come from similar work done on the cube, and apply to other puzzles beyond just the kilominx, but we want to collect what we know about the kilominx in one document.

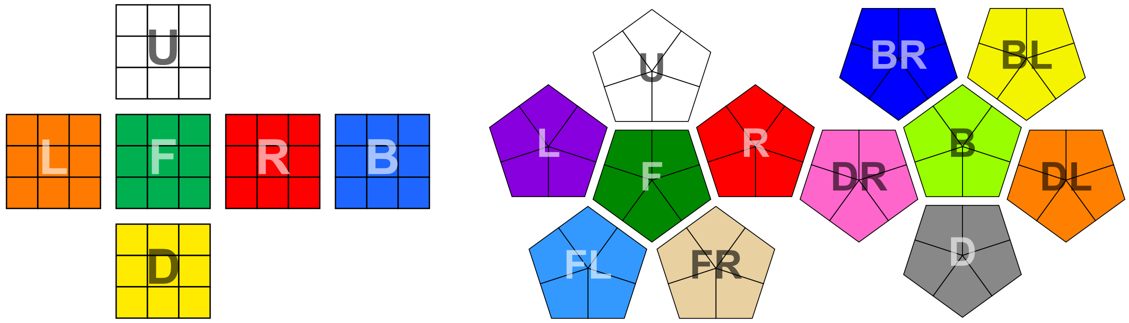
The kilominx is a twisty puzzle in the shape of a dodecahedron, with twenty corners, each with three stickers. Each face has a distinct color. This puzzle was originally introduced by Milton Bradley as the Impossiball in 1984 but in a spherical shape and with only six colors. In 2009 Meffert's started selling the Flowerminx which is in every relevant way a kilominx, but these days the puzzle is generally referred to as the kilominx.

The kilominx is usually sold with 12 distinct colors, one per face. If you use the Impossiball coloring, with only six colors, and opposite faces colored the same, there is no change in the mathematics of the puzzle. Even though there are two corners with white, red, and green stickers, one of the corners has them in clockwise order, and the other has them in counterclockwise order, so every corner is still distinguishable from every other, and the corners only combine in one way (modulo rotations) to make each face a solid color.



(a) Rubik's Cube

(b) kilominx



(a) Cube Notation

(b) Kilominx Notation

The kilominx is an excellent beginner puzzle. Despite having more states than the cube, the kilominx has only one type of piece (corners) while the Rubik's cube has two (corners and edges). Thus, the number of tricks (algorithms) you need is reduced; if you can figure out how to permute three corners, and how to twist two, you can solve the kilominx.

We will discuss the kilominx largely from a computational perspective, including bounds on God's number and other mathematical facts of interest, but there may be some content relevant to speedsolving. The kilominx is not currently an official puzzle in the World Cubing Association, unlike its bigger brother the megaminx. It is a popular puzzle nonetheless.

2 Notation

The notation for the kilominx follows the standard Singmaster notation for the cube[1], since both puzzles have moves defined by turning all the pieces (cubies) on a face. (The term cubie comes from the cube, where the individual small cubes that move are referred to by the diminutive; we retain the term even though the moving pieces of the kilominx are not cubic in shape. The term polytopie seems awkward.) We also adopt some conventions extending the Singmaster notation that are common in the speedsolving community. The faces are named with a sequence of upper-case letters. Moves start with a face name, but add move direction and twist amount defined by numeric suffixes and an apostrophe as an counterclockwise indicator. The faces of the cube are intuitive: U (up), D (down), R (right), L (left), F (front), and B (back). For the dodecahedral shape of the kilominx (and other minxes, that share the same face names), the naming is less intuitive. We name the white face U, the green face F, and the red face R, just like on the cube. With the standard coloring of the kilominx, the L face is purple, and instead of one more face adjacent to U, we have two, which we name BR (back right) and BL (back left). As on the cube, B is opposite F and D is opposite U. The remaining four faces are named FR, FL, DR, and DL, respectively.

Just as for the standard notation for the cube, a face name with no suffix is a minimal clockwise turn (90 degrees on the cube; 72 degrees on the kilominx). A lowercase version of a face name indicates a double-layer term (which on the kilominx moves 15 of the 20 corners). A suffix of 2 doubles the angle, and a suffix of an apostrophe makes the direction counterclockwise instead of clockwise. A suffix of v indicates a full puzzle rotation; we twist the face as indicated by the rest of the move, but instead of holding the rest of the puzzle fixed, we rotate the rest of the puzzle with the face. Occasionally, we will enclose a sequence in parentheses followed by a number to indicate repetition of the sequence that number of times.

Just as on the cube, many algorithms use only the U, F, R, and perhaps L faces, and many simple Rubik's cube algorithms work unchanged or with only slight changes. For instance, the algorithm F D' F' R' D' R U' R' D R F D F' U rotates two adjacent corners shared between the U and F faces on the cube. Changing D to FR yielding F FR' F' R' FR' R U' R' FR R F FR F' U has the same effect on the kilominx (and megaminx).

Our standard metric is the face-turn metric, where any rotation of any face counts as one move. An alternative metric is the *quantum-turn* metric, where only minimal face turns of 72° count as one move, and a 144° move counts as two moves.

3 The Group of the Kilominx

Like the cube, the state space of the kilominx forms a group. (This is not true, for instance, of the Rubik's Revenge, where some cubies are indistinguishable.) Each move cycles five cubies, which is an even permutation. Thus, only even permutations are reachable. In addition, the sequence U' F U R' U' F' U R permutes three corners, and with conjugations this permits us to permute any three corners, so every even permutation is reachable.

To understand corner orientations in twisty puzzles, usually an orientation convention is established. This involves marking one sticker of every cubie. For instance, on the Rubik's cube, the most common convention is that up and down stickers are marked, and for the middle edges, the front and back stickers are marked. You can then compute the twist of any cubie in any state by seeing how many clockwise rotations the marked sticker of the cubie is from the marked sticker in that place of the solved puzzle. To calculate the state space due to orientations on the kilominx, we do not need to evaluate any specific orientation convention. We need only observe that any particular move changes the twist of each cubie involved by a fixed amount. For example, the F move changes the twist of the cubie moving from the location intersecting the F, U, and R faces to the location intersecting the F, FR, and R faces by a fixed amount, regardless of the current identity or orientation of the cubie being moved. We can sum these numbers to get t, the total twist change for the move. We know that five twists of any face sum to the identity, so we know $5t = 0 \mod 3$. The only solution for this is if $t = 0 \mod 3$. Since this applies to all moves, this means that the sum of the twists of any state in any reachable position is always zero. (The same analysis applies to twists of corners on the Rubik's Cube.)

In addition, a previously given move sequence rotates one cubie 120° clockwise and another 120° counterclockwise. Combine this with permutations and we know that every orientation of cubies in any state is reachable, as long as the total twist sums to zero.

This gives us a state space of $(\frac{20!}{2})(\frac{3^{20}}{3})$ or 1,413,834,128,545,313,800,765,440,000 states (approximately 1.4×10^{27}). From a practical perspective, the orientation of the final puzzle does not matter, and there are no centers to fix that orientation as there are on the Rubik's cube, so you can group the states into equivalence classes by rotations: if a position can be rotated into another position, then the two positions are equivalent. If you do this, the state space is reduced by another factor of 60, which gives 23,563,902,142,421,896,679,424,000 or approximately 2.4×10^{25}. This state space is about a half million times the state space of the cube.

4 Full Puzzle Rotations

When calculating an optimal solution to a position, or computing God's number, or answering similar questions, it is important to account for rotational equivalence. To do this while still working in a group, you can hold a single cubie fixed (for instance, the cubie farthest from the F, U, and R faces) and treat the three moves that affect that corner differently. Instead of doing a move of five cubies, you can instead rotate the other fifteen cubies such that the fixed cubie remains fixed but the relative effect (modulo rotations) is the same as if that face had been twisted. Essentially, you replace generator B with the move f, which preserves the positioning of all cubies on the B face. You also do a similar replacement of the D and DL moves with u and l, respectively. In this smaller group, only 19 of the cubies move, and the state space is now $(\frac{19!}{2})(\frac{3^{19}}{3})$, precisely the number of equivalence classes with respect to rotations. Note that this transformation sacrifices most of the natural symmetry of the generators.

In the rest of this paper, when not clear from context, we will refer to the original group defined without symmetry reduction as the *larger group* and the reduced group as the *smaller group*.

5 Orders Of States

Given any sequence of moves on the kilominx, if we start from solved and repeat that sequence of moves some number of times, we will eventually return to the solved state. The minimum number of such repetitions is known as the order of that sequence. For instance, the sequence *U2* has order 5.

What orders are possible, and what is a shortest sequence that yields that order? In this section we work with the larger group.

To calculate this, we start by considering the possible orders of permutations of A_{20}, which is the subgroup of the kilominx group ignoring orientations. We can write any such permutation as a set of cycles. Since this is the alternating group, the count of cycles of even order has to be even. The order of such a permutation is just the least common multiple of the cycle sizes. For instance, if we have a permutation that consists of a cycle of size 5 and another cycle of size 7, this permutation will have order 35.

Adding the effect of orientations is easy. Any sequence of moves on the kilominx that does not permute the pieces does either nothing, or it twists two or more pieces. Applying the sequence three times will always yield the solved state. So when calculating the order of a sequence on the kilominx, we calculate each cycle of permutations of the cubies. If the sequence of moves for the pieces in that cycle has a total twist of 0 mod 3, then we use the cycle length in the least common multiple calculation, otherwise we use three times the cycle length.

Putting all this together, with a small program that enumerates the possible even cycle partitions on the permutations of 20 elements and accounting for orientation, we find 89 possible orders for sequences on the kilominx, with a minimum of 1 and a maximum of 630. The smallest order not possible is 23; no primes greater than 20 are possible. To obtain a cycle count of 630, we can use a move sequence that has a cycle of lengths 7, 5, and 6, at least one piece in the cycle of length 6 must be twisted when the sequence is repeated six times. One such move sequence is U D F2 DR2' DL2'. The most difficult orders to reach (in terms of the number of moves required) are 110, 234, and 378, each of which requires 7 moves. More details are in the repository.

6 Puzzle Rotations By Base Moves

Another question that arises (for the larger group) is whether the entire puzzle can be rotated using only face moves. Geometric shape-preserving rotations on the dodecahedron are rotations around an axis through two opposite corners, through the centers of two opposite edges, or through the centers of two opposite faces.

All such rotations are indeed possible. You can define an orientation convention that is preserved by the whole rotation around the centers of the U and D faces. Such a rotation consists of four 5-cycles and is thus even, so this full rotation is indeed possible. The other rotations through face centers can be obtained by conjugation by rotations (which essentially just permutes the moves), and rotations through edge centers and corners can be obtained by combining rotations through face centers. So, all such rotations are indeed possible legal states of the puzzle generated by just face twists.

Up to symmetry and mirroring equivalence, there are only four distinct rotations: a 72° twist around face centers, a 144° twist around face centers, a 120° twist through corners, and a 180° twist through edges. A face-centered 72° rotation requires at least 16 moves; one such sequence is F B2 DR2 FR2 FL2 F BR B' BL2' FR' BR2' R2' F2' B' BL FR'. In the quantum-turn metric, you can do it in 18 moves with F' FL L' DL BL' B BR' DR R' FR F' FL L' DL BL' B BR' DR. This sequence is interesting as it is equivalent to the easy to remember (F' FL Uv')9 but without the final rotation.

You can also do a 144° rotation around face centers in 16 moves, with U F2 U2' BR2' U2' L2 U2' D2 B2' FR2 DR2' FR2' R2' U2' B2 BL2. A rotation around corners is also possible in 16 moves with U' D BL' L2 BL2' DL L DR2' FR2 FL2' L2 DR' R' BR2 DR2' BR. We have not been able to find a short rotation around edges, but we know it requires at least 17 moves.

7 What Moves Generate The Group?

On the cube, it is not necessary to use all six face turns to attain all possible states; you can omit all turns of one face and still get to any position. This can be shown because the D move can done with F B L2 R2 F' B' U' F B L2 R2 F' B'. Surprisingly, this does not appear to increase the length of optimal solutions of random positions by more than a couple of moves, so some retail cube-solving robots hold the cube stationary and use only five motors to twist faces. On the kilominx, how many of the twelve faces turns must be included if we want to attain all possible states?

We can see that we do not need all the faces; the move U is equivalent to the sequence F BL' R2 BR' R BR2 R2' BR' R2 BR' R2 F' BL. At minimum we need to be able to move every cubie except one (which we can hold fixed). In addition, the face turns we use need to be connected to one another by shared cubies. Two faces that share a cubie share two cubies, and two connected faces can arbitrarily permute their 8 cubies within A_8 and also arbitrarily twist their cubies subject to the total twist being zero, using an argument similar to what we used for the full puzzle. So, all we need to do is find a connected set of pentagons on the dodecahedron that touch at least 19 corners.

The first face gives us five touched corners, and each subsequent connected face contributes at most an additional three. To reach 19 corners, we thus need $1 + \lceil (19-5)/3 \rceil$ or 6 total faces, and this can indeed reach all 20 cubies.

We can find a solution on a dodecahedron that touches all corners with only six connected faces, and this solution is unique up to symmetry. Indeed, the solution's complement is also a solution; it is a spiral shape that, for instance, consists of the cubies {FL, F, R, DR, B, BL} or its complement {FR, D, DL, L, U, BR}. Further, the perimeter of the solution is the single Hamiltonian path on the edges of the dodecahedron (up to symmetry).

If, instead of restricting yourself to face turns, you use deeper moves, then you can generate all positions with only two moves. The moves l and r (which, as you recall, move two layers rather than one) suffice. Solving the kilominx with only these two moves, while possible, is likely to be so challenging that we would love to hear if anyone accomplishes it without computer assistance. Every move affects 15 of the 20 corners; there is nowhere to hide! A random position, solved optimally, will likely take 42 or more moves. If we restrict ourselves to these two moves, we can do any minimal face turn optimally in 20 or 21 moves. For instance, to do U, we can use l2 r2 l2 r2 l2' r l2' r2' l2' r2' l2' r l2' r2 l2 r2 l2 r2' l' r2' (and this also works on the megaminx).

8 Canonical Sequences

How far from solved can a position on the kilominx can be? If we define the distance of a position as the fewest number of moves required to solve that position, then what is the maximum distance over all states of the puzzle? This is known as God's number.

In the face-turn metric, there are 12 faces and for each face there are four moves, so in total there are 48 possible moves. With $s = 2.4 \times 10^{25}$ states, an easy lower bound is just the smallest n such that $\sum_{0 \le i \le n} 48^i \ge s$; this gives 16 moves. Observing that there's no need to twist the same face twice consecutively does not lower the count. But we can do better.

Most pairs of moves on the kilominx commute; for instance, U and D commute, just as they do on the cube. You can do the moves in either order and the sum effect will be the same. Thus, any sequence contains a U followed immediately by a D will have the same effect as that sequence with those two moves swapped. We would like to account for this equivalence when calculating a lower bound for the states generated by sequences of a given length. Essentially, we want to identify a specific *canonical* sequence that stands in for all sequences that are equivalent up to exchanging the positions of adjacent moves that commute, or merging two moves on the same face.

We define a labeling function *unhidden* on a sequence of moves that returns another sequence of moves consisting of those moves in the original sequence that do not have a later move that does not commute with them. Next, we define a lexicographic ordering of the moves. With these we can build a finite state machine that accepts only canonical sequences. We use the return values of the *unhidden* function as states. A valid state is one where the *unhidden* function returns a sequence of moves where the last two moves are not on the same face (if the sequence has two or more moves) and the sequence is lexicographically ordered; otherwise, it is a rejecting state. Then, we define a canonical sequence as those in which all its prefixes are canonical sequences, and the return value of *unhidden* on the full sequence has moves in lexicographical order. To bootstrap, the empty sequence is a canonical sequence and the initial state of the finite state machine.

It is easy to use linear algebra to compute the count of accepting states in such a finite state machine. The results are shown in Table 1 (more data in the repository). Only when we get to a distance of 18 do we have enough total positions to cover all positions in the smaller group, so a lower bound on God's number for the Kilominx is 18 in the face turn metric. If we do the same analysis in the quantum-turn metric, we find a lower bound of 22 moves.

Length	Canonical Sequences	Positions
0	1	1
1	48	48
2	1,536	1,536
3	43,520	43,520
4	1,182,720	1,180,800
5	31,641,600	31,471,080
6	841,318,400	832,130,600
7	22,315,008,000	21,919,295,334
8	591,298,560,000	576,258,821,932
	...	
17	3.70×10^{24}	?
18	1.00×10^{26}	?

Table 1: Canonical Sequences and State Counts (face-turn metric). The linear recurrence for canonical states is $(36, -240, -320, 0)$ and the largest root of the polynomial is about 26.48, which is the asymptotic branching factor of canonical-sequence-based tree search.

This canonical sequence technique is not just useful in calculating a lower bound on God's number. It is also useful when exploring puzzles using graph search algorithms such as iterated depth-first search, because guiding the search with the canonical state finite state machine reduces the impact of encountering duplicated states during search. On the cube, a naive tree search has a ply of 18. Eliminating consecutive moves on the same face reduces this to 15. Using a canonical sequence state machine reduces this further to 13.3. On the kilominx the impact is more dramatic. A naive search has a ply of 48. Eliminating consecutive identical moves reduces this to 44. A canonical sequence state machine takes this down to 26.5.

9 State Counts by Distance

Another question that arises for twisty puzzles like the kilominx is the count of states at each distance. We can explore this by writing a simulation of the puzzle and exploring it with implicit graph search algorithms like breadth-first search or iterated depth-first search, making sure to account for states that might be reached by different paths.

We used the `twsearch`[2] program to perform such a search on the kilominx, both with the large group size and the small group size; the results are shown in Table 1. For the large group size, we were able to explore through a distance of 8, and these are the numbers shown. For the small group size, we were only able to explore through a distance of 7, and these numbers agreed exactly with the ones for the large group size, indicating that no two different states reachable through depth 7 are rotationally equivalent to each other. We also know that for the small group the count of states at distance 8 is a little smaller than for the large group, because the length 16 move sequence given earlier that rotates the full puzzle indicates that there is a state in the middle at distance 8 that occurs in two different rotations.

We look forward to extending this table as we apply more techniques to the problem Using a coset-based approach has a high likelihood of finding further results.

10 The Quantum Turn Metric

So far we have focused on the face-turn metric, but there is some interest in the quantum-turn metric as well. In this metric, we also explored canonical sequences and computed the count of positions at each distance (as far as our computational capacities extended). Those results are shown in Table 2.

In the quantum-turn metric, the lower bound on God's Number found using the canonical sequence analysis is 22.

Extending the canonical sequence finite state construction to work properly in the quantum-turn metric is straightforward; instead of forbidding duplicated faces in the last two moves, we forbid canceling moves and require that there be no more than two consecutive turns on a single face in

Length	Canonical Sequences	Positions (large group)	Positions (small group)
0	1	1	1
1	24	24	24
2	408	408	408
3	6,208	6,208	6,208
4	90,624	90,144	90,144
5	1,300,800	1,280,160	1,280,160
6	18,538,560	179,688,00	17,968,800
7	263,393,280	250,521,744	250,521,744
8	3,737,257,920	3,478,019,136	3,478,019,136
9	52,996,688,000	48,146,093,162	48,146,093,102
10	751,335,964,800	665,042,912,002	?
		...	
21	3.49×10^{24}	?	?
22	4.94×10^{25}	?	?

Table 2: Canonical Sequences and State Counts (quantum-turn metric). The linear recurrence for canonical states is $(18, -42, -160, -180, -120, -40, 0)$, and the largest root of the polynomial is about 14.17.

the same direction. We also perform some additional small improvements in our finite state machine construction, such as identifying states that have the same effect and combining them.

11 Optimal Solving

Optimal solvers for the Rubik's Cube have existed since 1997, when Richard Korf[3] optimally solved ten positions. He used iterated depth-first search along with a *pattern database* built from states near solved that allowed his search to backtrack when it was clear a particular search subtree would not yield a short solution. Presently, optimal solvers exist that can solve random positions of the cube in well under a second each. The state space of the kilominx is only 2.4×10^{25}, not too much larger than the 4.3×10^{19} of the Rubik's Cube, right? And technology has come a long way in 19 years. Surely we can optimally solve positions of the kilominx?

We tried. We used a machine with 1.5TB of memory and 40 threads on 20 physical cores. We implemented iterated depth-first search, using a huge 1.4TB pattern database, and fed it some random kilominx positions. After about two days it found that the first position had no solution in 17 moves, but another solid week of computation did not suffice to find a solution in 18 moves or show that 18 moves was not sufficient. We believe we are close to being able to optimally solve random positions on the kilominx, but it may be another few years or require better programmers.

But all is not lost.

12 Two-Phase Solvers

In 1992, well before optimal solvers for the cube existed, Herbert Kociemba extended some earlier group theory ideas and created a program that quickly generated nearly-optimal solutions to the Rubik's Cube[4]. The efficiency of this *two-phase* program was stunning; before then, finding short solutions was nearly impossible. But Kociemba's two-phase idea generated solutions within a move or two of optimal, using very little memory and very little computer time.

The way it worked was to first bring the position to be solved into a particular subgroup of the cube group, and then solve it within that cube group. Each step of this algorithm was executed using iterated depth-first search and pattern databases (just like Korf's 1997 solver), but because each subproblem was much simpler, solutions could be found very quickly. The key idea was to find many sequences that brought the position into the relevant subgroup, and evaluate all of them for short solutions in the second phase.

Subgroup	Phase 1 size	Phase 1 avg	Phase 2 size	Phase 2 avg	Expected
{U, R}	5.34×10^{17}	12.97	4.41×10^{7}	12.26	25.23
{U, R, F}	6.60×10^{14}	11.11	3.57×10^{10}	11.76	22.87
{U, R, F, L}	5.55×10^{11}	9.11	4.24×10^{13}	13.43	22.54
{U, R, F, L, BR}	3.39×10^{8}	6.97	6.95×10^{16}	15.52	22.49
{U, R, F, L, BR, BL}	7.53×10^{6}	5.89	3.13×10^{18}	15.77	21.65

Table 3: Two-phase approaches for the kilominx considering only one phase-one solution.

A similar idea works well for the kilominx. For example, we can use the subgroup {U, F, L, R} as the target; this subgroup has size 4.2×10^{13} and its coset graph has size 5.6×10^{11}, so iterated depth-first search with a pattern database can easily find optimal solutions for each phase. The combined solutions are not guaranteed to be optimal, but they are not too far off.

We can get a lower bound for the average optimal solution length for random positions of the small group using the canonical sequence approach from an earlier section. For the Kilominx, this value is 17.83 moves. We expect the actual average to be higher than this, but without an optimal solver we cannot put a number on it. For comparison, for the two-phase approach we can approximate the solution lengths from each of the individual phases by solving a bunch of random positions. We used 50,000 positions. The results are in Table 3. We can see a simple two-phase approach, even considering only a single phase-one solution, will usually get us within five moves of optimal.

Using Kociemba's trick of finding many solutions for phase 1 and evaluating each for the shortest possible solution in phase 2, we can get closer to optimal. To evaluate many solutions can be much faster than might at first appear, since we limit the phase 2 search to only find solutions that are better than the ones we have so far, so we do not need to search as deeply in phase 2. This leads us to prefer a phase 2 that has more states than our phase 1, so we can concentrate on finding many phase 1 solutions quickly. By considering the first 1,000 phase 1 solutions with the U R F L BR BL subgroup on a set of 10,000 random positions, we observed an average solution length of 20.5 moves.

As a further optimization, we can also consider all rotations, mirroring, and also the inverse positions, since all of these are just conjugations by rotations or sequence inversions, neither of which changes the sequence length in our metrics. On the kilominx this gives us 240 initial possibilities, some of which will likely have shorter solutions in our two-phase approach than others. While we did not perform the experiment, we expect this idea of using multiple initial seeds would reduce the length of found solutions by about a move.

13 Upper Bounds on God's Number (31 and 42)

Earlier, we calculated a lower bounds on God's number for the small group of 18 in the face-turn metric and 22 moves in the quantum-turn metric. We now consider upper bounds. As earlier, for God's number we consider the smaller group. We use the usual approach, which is a chain of subgroups.

First, we reduce the puzzle to the subgroup {U, F, R, L, BR, BL}. This step has 7,534,944 cosets and can be done in 7 moves in the half-turn metric and 10 moves in the quantum-turn metric. Next, we reduce the puzzle to the subgroup {U, F, R} using only the moves in the prior subgroup. This step has 87,567,480 cosets and can be done in 10 moves in the half-turn metric and 14 moves in the quantum-turn metric. Our final step is to solve the rest of the puzzle within the subgroup {U, F, R}. This analysis was originally carried out by Ben Whitmore[5] and verified by us. In the half-turn metric, this final step takes 14 moves, and in the quantum-turn metric it takes 19 moves.

With this subgroup chain, we establish that every position on the kilominx can be solved in 7+10+14=31 moves in the face-turn metric, and 10+14+19=43 moves in the quantum-turn metric.

In the quantum-turn metric, we can improve these results. The {U,F,R} subgroup has only 2,243 positions at a distance of 19. We optimally solve these positions using {U,F,R,L,BR} and find all can be solved in 18 or fewer moves. This reduces the bound by one move, to 42 moves in the quantum-turn metric.

moves	one look		perm (safe)		perm (unsafe)		ori (safe)		ori (unsafe)	
{U, F}	11.18	14	9.35	13	11.75	14	7.13	9	8.49	11
{U, F, R}	9.62	12	8.45	12	10.27	11	6.52	8	7.23	9
{U, F, R, L}	9.27	12	8.15	10	10.00	11	6.35	8	7.09	9
{U, F, R, L, BR}	9.11	11	8.05	10	9.90	11	6.25	7	6.99	9
{U, F, R, L, BR, BL}	8.95	11	7.98	10	9.88	11	6.23	7	6.91	9
(all)	8.92	11	7.98	10	9.88	11	6.23	7	6.91	9

Table 4: Last layer average move count in face-turn metric.

moves	one look		perm (safe)		perm (unsafe)		ori (safe)		ori (unsafe)	
{U, F}	15.67	19	14.15	18	14.96	18	9.17	12	11.54	15
{U, F, R}	12.39	16	11.70	14	12.49	14	7.45	10	9.09	12
{U, F, R, L}	11.53	15	10.58	13	11.68	13	7.03	9	8.67	12
{U, F, R, L, BR}	11.21	14	10.37	12	11.43	13	6.87	8	8.47	11
{U, F, R, L, BR, BL}	10.92	14	10.27	12	11.23	13	6.77	8	8.40	11
(all)	10.80	14	10.27	12	11.23	13	6.77	8	8.40	11

Table 5: Last layer average move count in quantum-turn metric.

14 Last Layer Analysis

When solving the kilominx by hand, one approach is to solve it layer by layer. First we solve one face (the bottom, often the white face), then the five cubies adjacent to that face, then the five cubies adjacent to those five, leaving only the top face to be solved. Each of these steps except the last can be done quickly and intuitively. But just like on the cube, solving the last layer quickly can require memorized algorithms.

One approach is to solve the permutation of the last layer first, and then the orientation; alternatively, we can solve the orientation and then the permutation. Advanced solvers might memorize enough algorithms to solve both simultaneously. (In theory, there are $(\frac{5!}{2})(3^5/3)$ or 4,860 relevant positions, but these can be reduced by symmetry, mirroring, inversion, and a final last-layer twist (AUF) to 196—still a daunting number.)

To computers, 4,860 is a small number, so we solved all last-layer positions, all last layer permutations both preserving (safe) and not preserving (unsafe) orientation, and all last layer orientations both preserving and not preserving permutations. We compute average and maximum move counts for all positions using two, three, four, five, six, and all faces (preferring the faces in the following order: U, F, R, L, BR, BL, and then the rest). Table 4 summarizes the expected and maximum move counts for the face-turn metric; Table 5 does the same for the quantum-turn metric.

If you solve permutation before orientation, you will choose the unsafe permutation algorithms followed by the safe orientation algorithms. If you solve orientation first, then you will choose the unsafe orientation algorithms followed by the safe permutation algorithms. From a recognition perspective, doing orientation first makes the most sense. This also has a slight move count advantage.

15 Future Opportunities

The computational results in this work used the `twsearch` program, which has been written generally to support many different puzzles. We expect code written specifically for the kilominx to likely be much faster, just as optimal solvers for the cube are much faster than using `twsearch` to find optimal cube solutions, as both low-level move routines can be carefully optimized and high-level characteristics of the specific puzzle can be exploited. Thus, for instance, it is likely that an optimal solver for the kilominx that is reasonably effective can be written, and it is certain that an efficient two-phase solver can be written. We have not done so (yet), but we see no technical impediment.

To extend some of the more theoretical results, such as the count of positions at each distance and better bounds on God's number, coset approaches can likely be exploited, as was done on the cube.

We estimate that our abilities on the kilominx currently correspond approximately to where we were in 1990 on the cube. We look forward to continued progress in exploring this and other twisty puzzles.

16 Additional Data and Code

For space reasons, we have omitted most numerical data in this report, and instead put our code and data at https://github.com/rokicki/kilominx. The code depends on the `twsearch` program.

References

[1] Singmaster, David. *Notes on Rubik's Magic Cube*, 5th edition. London: Mathematical Sciences and Computing, Polytechnic of the South Bank, 1980.

[2] Rokicki, Tomas, *twsearch*, https://github.com/rokicki/twsearch.

[3] Korf, Richard E. "Finding Optimal Solutions to Rubik's Cube Using Pattern Databases." *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, Providence, RI, July, 1997, pp. 700–705.

[4] Kociemba, Herbert. "Close to God's Algorithm" *Cubism For Fun* 28 (April 1992), pp. 10–13.

[5] Whitmore, Ben, *Kilominx can be solved in 34 moves*, http://forum.cubeman.org/?q=node/view/560