IA Group 5

Math AA HL

Title:

Investigations into Sliding Puzzles

RQ:

What are the solve conditions, fastest solving methods, and generalisability of the 15-puzzle?

Page count: 20

Candidate Code: HYK028

May 2021

1 Introduction

In the 1870s, the 15-puzzle was created by Noyes Chapman, a postmaster in New York [SW20], with 15 tiles, numbered 1-15, placed on a 4×4 grid leaving the final space blank.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Figure 1: Solved state of the 15-puzzle

1	2	3	4
5	6	7	8
9	10	11	12
13	15	14	

Figure 2: 14-15 puzzle scramble

The puzzle is scrambled by sliding the pieces around, and the objective of the puzzle is to slide the pieces back into the solved state. A piece can only be moved into an adjacent blank space, and the entire puzzle cannot be rotated (orientation is maintained). The puzzle gained rampant popularity in 1880 because of a special scramble case known as the 14-15 puzzle [SS06]. The scramble swapped the 14 and 15 tiles from the canonical state, and the aim was to then solve the puzzle. Interestingly, the swap of the tiles rendered the puzzle unsolvable.

This lead to the simple question of why was it unsolvable? Furthermore, are there any other scrambles which are unsolvable? This became a significant issue when I attempted to create a computer version of the 15-puzzle. The computer allowed me to track how fast I was solving the puzzle, which allowed me to race against friends. The problem was generating the scrambles, because about half of the scrambles I generated at random were unsolvable. I could manually check if a scramble was solvable by running a breadth-first search with my computer, but that would be time consuming. It would also be a thwarted opportunity to apply mathematics to the problem. I wanted to find a quick way to check if a scramble was solvable, maybe even quick enough for a human to do. This formed the core of my exploration.

Of course, as I kept playing with the puzzle, and exploring the mathematics, I became interested in other questions as well: What if the puzzle had different dimensions? What is the fastest way to solve the puzzle, according to mathematics? Hence, the exploration grew to have three aims: Firstly, to determine the set of all solvable scrambles; secondly, to investigate the generalisability of any conclusions from the 15-puzzle to all slidy puzzles; and lastly, to statistically determine the fastest method to solve the puzzle.

2 Solvability

A typical approach to analysing a puzzle is identifying an invariant. An invariant is a property of the puzzle which remains unchanged under a given class of transformations [Wei20]. Simply speaking, moving a piece into an adjacent empty space is the transformation for a 15-puzzle. Hence an invariant would be a property of the puzzle which always stays the same regardless of the number of moves made. However, trying to create a mathematical definition for the invariant is difficult. Sliding a piece into an adjacent space hasn't yet been mathematically defined.

2.1 Permutations

The appropriate mathematical structure I identified was the permutation of a set. A permutation is a bijective function from a set to itself [QHH13]. A bijective function has the properties of injectivity and surjectivity:

1. **Injectivity:** Each element of the domain set is mapped to a unique element of the range set (where the domain is the input, and the range is the output).
2. **Surjectivity:** Each element from the codomain set has a corresponding element in the domain set (codomain is the set of all possible values of the function).

A bijective function maps every element of one set to exactly one element of the second set. Hence, a permutation, f, maps every element of a set back to an element in the same set. It uses the following notation for a set of size n, $\{1, 2, \cdots, n-1, n\}$:

$$f = \begin{pmatrix} 1 & 2 & \cdots & n-1 & n \\ f(1) & f(2) & \cdots & f(n-1) & f(n) \end{pmatrix}$$

The function can be represented as a permutation in this form. For an example, consider p:

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & 1 & 6 & 5 & 4 \end{pmatrix}$$

This permutation sends 1 to 2, 2 to 3, 3 to 1, 4 to 6, 5 to 5, and 6 to 4. Permutations can also be applied one after another, known as compositions. They are read from right to left in a composition. A permutation $g = pqr$ would be evaluated by applying r, then q and then p. Compositions of permutations are not commutative, hence these cannot be swapped. However, the compositions are associative, allowing qr to be evaluated and then p OR r and then pq.

This brings us to cycle notation, which represents the permutation as a composition of cycles of elements. In the above example, p, 1 goes to 2, 2 goes to 3, and 3 goes back to 1.

Hence, there is a cycle: $1 \mapsto 2 \mapsto 3 \mapsto 1$. This cycle can be represented as $(1\ 2\ 3)$. If all the cycles are taken into account:

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & 1 & 6 & 5 & 4 \end{pmatrix} = (1\ 2\ 3)(4\ 6)(5) = (1\ 2\ 3)(4\ 6)$$

Each cycle is composed together and read from right to left. Note that any cycle with only one element is removed since it maps back to itself. Usually, cycle notation is used since it is shorter and easier to calculate compositions by hand. However, in the context of the 15-puzzle problem, it helps to decompose the permutation into transpositions.

2.2 The 15-puzzle

The parallel between the permutations on a set and the 15-puzzle become clear quickly. The permutation is an individual move, where a piece changes position. The set is every possible position on the puzzle. A valid scramble should be able to be represented as a composition of permutations (a list of moves). The first step to mathematically defining the 15-puzzle is to decide upon a valid set of positions:

$$P = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16\}$$

This set of positions is represented on the puzzle below:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

There are only 15 tiles on the puzzle, but there are 16 positions, hence the 16-th position is shown. It is important to note the distinction between a tile and a position. Any given position on the puzzle may include any of the tiles, or have no tile. Since the difference between a tile and a position can get confusing, the positions are labelled as numbers, but the tiles are labelled as letters, an idea borrowed from a paper at Louisiana State University [Cha+]:

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	

Each move on a 15-puzzle is a swap between two of the positions indicated above, where the contents are exchanged; the contents being tiles. This can be represented as such:

$$\begin{pmatrix} 1 & 2 & \cdots & n & \cdots & m & \cdots & 15 & 16 \\ 1 & 2 & \cdots & m & \cdots & n & \cdots & 15 & 16 \end{pmatrix}$$

Each of these swaps are between exactly two positions, and the mathematical term for this is a transposition. In cycle notation, this is $(n\ m)$. The uniqueness of the 15-puzzle is that not every transposition is an actual move. Moves on the 15-puzzle require the positions to be adjacent to one another, and also that a blank tile is available in either of those positions. Hence, there needs to be a better definition of different types of transpositions.

Let us consider some possible transpositions. Starting with the solved state, the permutation (12 16) is quite simple. It involves swapping the L piece and the blank tile. A simple move like this can be called a trivial transposition. We can call this permutation $S_{12,16}$.

A more complicated transposition is swapping the contents of 7 and 16, which moves the

G tile into the blank tile. These positions are not adjacent, and hence the transposition is called non-trivial. It is called $S_{7,16}$. Of course, this needs to be represented as a series of actual moves. In layman's terms, it is a sequence of 11 moves of the blank tile: L, U, U, R, D, L, D, R, U, U, L (R - right, L - left, U - up, D - down). In permutation notation (reading from right to left):

$$
\begin{aligned}
S_{7,16} &= (8\ 7)(12\ 8)(16\ 12)(15\ 16)(11\ 15)(12\ 11)(8\ 12)(7\ 8)(11\ 7)(15\ 11)(16\ 15) \\
&= (7\ 16)(8)(11)(12)(15) \\
&= (7\ 16)
\end{aligned}
$$

In this example, $S_{7,16}$, which is a non-trivial transposition, is broken down into a list of moves, which are trivial transpositions. Non-trivial transpositions are actually just a special type of scramble, where only two positions have been swapped with one another. However, they are quite useful since any valid scramble can simply be written out as a series of non-trivial transpositions. Consider the semi-complex scramble:

B	A	C	D
E	F	H	G
I	J	K	L
M		N	O

This scramble has a list of changes:

1. (1 2): A $\mapsto$ B $\mapsto$ A
2. (7 8): G $\mapsto$ H $\mapsto$ G

3. (15 14)(16 15) or (15 16 14): Blank $\mapsto$ O $\mapsto$ N $\mapsto$ Blank

These are all non-trivial transpositions, or compositions of non-trivial transpositions. However, to actually execute the moves to get to this scramble, the non-trivial transpositions need to be written out in terms of trivial transpositions (actual executable moves). This is the part of the question of solvability. Can the scramble be broken down into a list of trivial transpositions? And if so, can these trivial transpositions actually be executed on the puzzle?

2.3 The Parity Theorem

The above scramble happens to be valid, as I checked with an actual 15-puzzle. However, the real challenge is showing this in a simple, fast mathematical way, rather than listing a long sequence of moves that can create this scramble. At this point, a mathematical formalism has been identified, but some more tools are necessary to arrive at a conclusion.

The invariant is a property of the puzzle which stays the same regardless of the number of moves made. To find such a property, a connection must be made between a scramble, its non-trivial transpositions (swaps) and its trivial transpositions (actual moves). This connection will allow the scramble to be tied back to the original puzzle. The appropriate tool in this scenario would be the parity theorem.

Theorem 1. *If S can be expressed as two unique compositions of transpositions:* $S = \tau_1 \circ \tau_2 \circ \cdots \circ \tau_k = \beta_1 \circ \beta_2 \circ \cdots \circ \beta_m$, *then* $k + m \equiv 0 \pmod 2$

This theorem states that if S has two possible compositions, then the number of transpositions in both possible compositions must both be even or both be odd, in order for their sum to be even. Since any scramble can be decomposed into non-trivial transpositions, and also decomposed into trivial transpositions, the theorem is an incredibly useful tool to compare the two.

However, before proving the theorem, it is important to introduce some more concepts with reference to permutations.

2.3.1 Identity & Inverse

ϵ is the identity permutation. It has no impact on a permutation when applied in composition. In cycle notation, it has no written form, but in original permutation notation:

$$\epsilon = \begin{pmatrix} 1 & 2 & \cdots & n \\ 1 & 2 & \cdots & n \end{pmatrix}$$

An inverse is essentially an opposite permutation. Earlier, the permutation was defined as a bijective function which maps elements of a set to itself. The inverse of a permutation maps the elements of the codomain to the elements of the domain. This is a numerical example:

$$B = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 1 & 5 & 4 \end{pmatrix} \to \begin{pmatrix} 2 & 3 & 1 & 5 & 4 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix} \to B^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 1 & 2 & 5 & 4 \end{pmatrix}$$

The first step flips the domain and codomain of the permutation, which is taking the actual inverse. This could be left as is, but in permutation notation, the domain must be ordered. This leaves the final notation for B^{-1}. This happens to be an easier process with cycle notation:

$$B = (1\ 2\ 3)(4\ 5) \to B^{-1} = (5\ 4)(3\ 2\ 1)$$

The order and content of the cycles are reversed. Note that reversing the contents of

(4 5) to (5 4) doesn't actually change the cycle. 4 still goes to 5, and 5 still goes to 4. Hence, the contents being reversed is not a concern with 2-element cycles (transpositions). Any permutation written as a composition of transpositions can simply be written in reverse.

Unsurprisingly, if a permutation and its inverse are both applied to a set, no changes are made, producing the identical result as the identity permutation.

2.3.2 Signature

Every permutation has a classification as either even or odd, depending on the number of transpositions present in the permutation. Its state of being even or odd is known as the signature: 1 for even, -1 for odd. The parity theorem states that any permutation must always be even or odd, regardless of how it is represented in terms of transpositions. However, before proving the theorem, the identity permutation must be shown to always be even.

Based on my initial intuition, it did seem obvious that the identity permutation must be even. Any applied transposition must eventually be reversed, which results in an even number of transpositions. However, I could see how this result would be disputed and not obvious if there were more complicated cycles. Hence, a rigorous mathematical methodology is necessary. The key to showing this result is to formalise a permutation based on its Vandermonde polynomial and associated signature. These definitions of the permutation allow for a more rigorous proof.

The Vandermonde polynomial is defined as [Dun08]:

$$P(x_1, \ldots, x_n) = \prod_{i<j}^{n} (x_i - x_j)$$

for a permutation on a set with n unique elements. Any transposition of elements, u and v, for example, will swap every instance of x_u and x_v in the polynomial. For example:

$$P(x_1, x_2, x_3) = (x_1 - x_2)(x_2 - x_3)(x_1 - x_3)$$

If a relatively simple permutation were applied, such as σ=(1 2), which swaps 1 and 2 [Bas+08]:

$$P(x_{\sigma(1)}, x_{\sigma(2)}, x_{\sigma(3)}) = (x_2 - x_1)(x_1 - x_3)(x_2 - x_3)$$

The signature of a permutation is given by the polynomial of the transformed permutation (permutation with the attached transpositions) divided by the identity permutation polynomial:

$$\operatorname{sgn}(\sigma) = \frac{P(x_{\sigma(1)}, \ldots, x_{\sigma(n)})}{P(x_1, \ldots, x_n)}$$

Let us look at the actual value of the signature of σ:

$$\operatorname{sgn}(\sigma) = \frac{P(x_{\sigma(1)}, x_{\sigma(2)}, x_{\sigma(3)})}{P(x_1, x_2, x_3)} = \frac{(x_2 - x_1)(x_1 - x_3)(x_2 - x_3)}{(x_1 - x_2)(x_2 - x_3)(x_1 - x_3)} = -1$$

Most factors stay the same, with some slight reorganisation, but the swap from $(x_1 - x_2)$ to $(x_2 - x_1)$ causes a flip in the signature to negative. Hence, a single transposition will flip the signature from 1 (even) to -1 (odd). It is simple to see that with every added transposition, the sign flips. Hence, for σ with m transposition $\operatorname{sgn}(\sigma) = (-1)^m$.

ϵ, by definition, must not impact the signature of a permutation, since it returns the same permutation. Hence, $\operatorname{sgn}(\epsilon) = 1$. This implies that the number of transpositions, m, in ϵ, must be even. The identity is even.

2.3.3 Proof

The above information can now be implemented. Consider the two expressions of S again:

$$S = \tau_1 \circ \tau_2 \circ \cdots \circ \tau_k = \beta_1 \circ \beta_2 \circ \cdots \circ \beta_m$$

Consider the value of ϵ in terms of the two expressions of the transpositions:

$$
\begin{aligned}
\epsilon &= S \circ S^{-1} \\
&= (\tau_1 \circ \tau_2 \circ \cdots \circ \tau_k) \circ (\beta_1 \circ \beta_2 \circ \cdots \circ \beta_m)^{-1} \\
&= (\tau_1 \circ \tau_2 \circ \cdots \circ \tau_k) \circ (\beta_m \circ \beta_{m-1} \circ \cdots \circ \beta_1)
\end{aligned}
$$

ϵ is expressed as a composition of $k + m$ transpositions. ϵ can only be expressed as an even number of transpositions, which indicates that $k + m \equiv 0 \pmod 2$. This proves the parity theorem, any two expressions of a permutation must always be even, or always be odd, to ensure $k + m$ is even.

2.4 Invariant

The final step of looking at the solvability is identifying the invariant. Naturally, this is a step involving creativity and thought. It took a few hours of trying several things and pondering the meaning behind the puzzle's moves. Of course, given my understanding of permutations, I had some intuition for what techniques would be useful.

The invariant I noticed was that if the number of moves are even, the taxi-cab distance (shortest path along the positions) of the blank tile from the solved position will also be even, and the same if they were both odd. This is obvious from looking at each move being made, which moves the blank tile away from or closer to the solved state by exactly 1 position. Recalling that moves are represented by trivial transpositions:

$$
(\text{\# of trivial transpositions} + \text{taxi-cab distance}) \equiv 0 \pmod 2
$$

This ties together perfectly with the parity theorem. Let's say we have a list of non-trivial transpositions, which are relatively easy to find, as was shown above. By the parity theorem, the number of non-trivial transpositions must be equal to the number of trivial transpositions, modulo 2. Hence, the number of non-trivial transpositions must be equal to the taxi-cab distance of the blank tile from the solved state, modulo 2. Both of these are relatively quick to calculate. If this condition isn't satisfied, the scramble is not solvable. This solution of using the invariant is a closed form solution for a computer to check if a given scramble is solvable, and whether it should be given to a player of the 15-puzzle.

An interesting observation I made was that exactly half of all scrambles are solvable [Gra02], since all of them must be categorised as having an even invariant or an odd invariant (where the invariant is the sum of the number of trivial transpositions and the taxi-cab distance).

2.5 Application and Extension

These conclusions can then be applied to the 14-15 puzzle posited in the 19th century. The problem is shown in it's original notation and the adjusted letter notation:

1	2	3	4
5	6	7	8
9	10	11	12
13	15	14	

Figure 3: 14-15 scramble

A	B	C	D
E	F	G	H
I	J	K	L
M	O	N	

Figure 4: 14-15 scramble in letter notation

This scramble has only one non-trivial transposition: (14 15). By the parity theorem, the number of moves required to reach this state must also be odd, and the blank square must be an odd number of tiles from the solve state. However, the blank square is 0 tiles away from the solved state, which is even. Hence, this scramble is impossible to solve.

While I had solved the original problem, I was curious about the solvability of other types of sliding puzzles. The puzzle could be extended to have larger dimensions, or even more dimensions. The 15-puzzle is a 4×4 puzzle with a length of 4 tiles and 2 dimensions. A 3-dimensional puzzle would be a cube, and higher dimensional puzzles would be more abstract problems. The question I asked was: what is the solvability of a puzzle with n tiles and k dimensions? What about a non-square sliding puzzle, perhaps a rectangle or cuboid? It turns out that the mathematical structure of a permutation is quite versatile. A puzzle with k dimensions, with a respective dimension length of x_i will have a set of positions, P, where:

$$|P| = \prod_{i=1}^{k} x_i$$

$|P|$ represents the size of the set of positions. As long as the set can be constructed and the definition of a move remains the same (swapping the blank tile with an adjacent tile), the same logic can be used with an arbitrarily large set of positions. The non-trivial transpositions can be found and compared with the taxi-cab distance. Hence, for any sliding puzzle with $x_i > 1$ and $k > 1$, exactly half of the states are solvable. I introduce bounds since $1 \times x$ puzzles clearly operate differently. They have swaps along a single line, which is not consistent with most of our work so far.

3 Optimisation

The final part of this investigation focuses on finding the fastest way to solve the 15-puzzle, a problem I faced when attempting to race with friends. For the purpose of this investigation, the “fastest” way to solve the puzzle is the method with the least number of moves. This may be different in practice, since certain combinations of moves can be executed faster.

3.1 Group Theory

I had recently studied elementary group theory and its applications to puzzles such as the Rubik’s cube. Then, as I worked through my investigation, I noticed parallels between group theory and permutations. Hence, I was optimistic that the structures in group theory would help with the optimisation problem for the 15-puzzle.

A group is a set defined upon a binary operation [Red19]. The binary operation may use any two elements of the set to produce a new element, that must be within the set itself (by definition). The first step to apply group theory, is to define the group itself:

1. **Set:** It may be tempting to use the same set of positions used for the permutations, but this actually would not be able to produce a group. A new set must be created, which is the set of all permutations of the puzzle. More specifically, all the valid scrambles of the puzzle, which would include every permutation with an even parity.

2. **Operation:** Since the set is all permutations of the puzzle, the probable operation would be permutation composition. It uses two permutations to create a new permutation, and it is the only known operation which works as necessary.

These were the choices for a set and an operation which I had made at first observation. However, the choice of set is actually quite problematic. Two scrambles cannot be composed, since the blank tile ends up at different positions after each scramble. Any scramble needs the blank tile to be starting at position 16. Hence, any given scramble should be modified to move the blank tile into position 16, so it can actually be composed.

The effect that this has is that every single scramble becomes an even permutation. If a permutation were already even, the parity states that the blank tile would be an even number of spaces away from position 16. To return the blank tile to position 16, it would take an even number of moves or transpositions. The sum of transpositions turns out to be even. If a permutation were odd, the parity states that the blank tile would be an odd number of spaces away from position 16. To return the blank tile to position 16, it would take an odd number of moves or transpositions. The sum of transposition is, once again, even. Since any scramble with an even parity can be transformed into an even permutation, the set can be rewritten as the set of all even permutations.

However, I needed to prove it was actually a group, which must meet 4 conditions:

1. **Closure:** The product of any two elements in a composition must be within the set of even permutations.

   When any two permutations are composed, it is the equivalent of combining all their transpositions. Since both elements involved are even permutations, they both have an even number of transpositions. When combined, they must produce a permutation with an even number of transpositions as well. Hence, the group is closed.

2. **Identity:** A group must have an identity. An identity permutation has already been identified.

3. **Inverse:** A group must have an inverse. Any even permutation has already been shown to possess an inverse, which is also an even permutation.

4. **Associativity:** This simply means that a series of permutations can be composed in any order, which has also been shown earlier.

Hence, the set of even permutations defined on the binary operation of composition is a group. I initially thought that identifying this group would help with finding the fastest way to solve the 15-puzzle. I would create a graph of every single permutation, and connections between each one, creating a pathway along every possible permutation. This would make it a graph theory problem, which I thought would be simpler to solve.

I even thought that the group notation would actually be useful to formalise a solution. However, my efforts were quite futile. To use a computer with graph or group notation to solve a problem would involve an immense amount of brute force and sorting through up to $\frac{16!}{2} = 1.05 \times 10^{13}$ permutations in the group. From a pure mathematics perspective, I had no tools to even begin a group theory analysis.

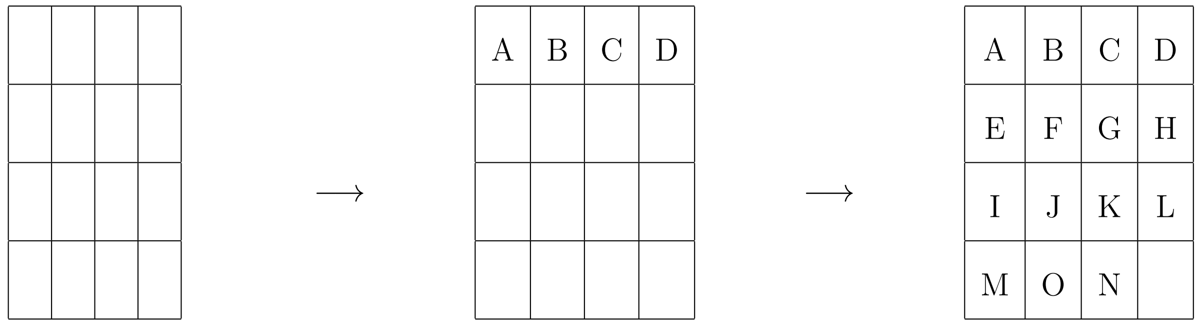
However, I realised that this is an inevitability of mathematical exploration. Often, I must try multiple approaches to a problem to arrive at a final solution, even in tests. Most of the approaches tend to be incorrect, but the process itself is necessity in reaching the solution. Hence, I am not disheartened, but simply more aware of the possible approaches in analysis of the 15-puzzle.

3.2 Solve Techniques

At this point, I did want to find an actual technique to solve the puzzle at the least. Hence, I attempted to find the fastest human technique to solve the puzzle. A computer can follow the technique (which is an algorithm) and output the best possible way to solve a scramble using a given technique. In my experience speed-solving the 15-puzzle, I used two such techniques: layer-by-layer (LBL) and fringe. Both variations of the technique I am using are an advanced version. This means that many steps in the technique are skipped and the solver intelligently places pieces.

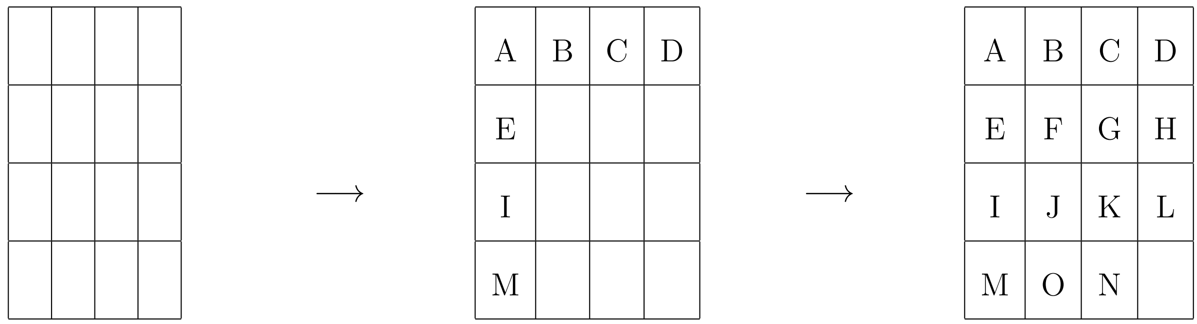
LBL Advanced

This technique solves the puzzle layer by layer, more specifically solving 2×2 blocks in an order. The beginners variant solves the puzzle in this order: AB, CD, EF, GH, IM, JKLON. An experienced solver will use: ABCD, EFGHIJKLMON. This simultaneously solves all pieces in the first row and then all the leftover pieces.



Fringe Advanced

This technique solves the puzzle by fringes. The beginners variant solves the puzzle in this order: AB, CD, EIM, FGH, JO, KLN. An experienced solver will use: ABCDEIM, FGHJOKLN.



3.3 Statistical Analysis

It is assumed in the speed-solving community that the fringe technique is usually a faster method to solve the 15-puzzle, but I was sceptical of this notion. Hence, instead of picking one technique and finding the average move-count, I decided to compare a sample of 1000 solves for each technique. Assume that the move-count for fringe solves follows a normal distribution of F, and LBL solves follow a normal distribution of L. The central limit theorem allows us to make this assumption. Let us conduct a one-tailed test.

$$H_0 : u_f - u_l = 0$$

$$H_1 : u_f - u_l < 0$$

The null hypothesis considers that the two techniques are roughly equal in speed. The alternate hypothesis is that the fringe solving is actually faster. I will use a program called SlidySim developed by "ben1996123" on a speed-solving forum. The program solves randomised scrambles using a given technique and outputs the move-count for each solve. The results of solving 1000 scrambles with each technique are shown below. The data-set is too large to include even in an appendix.

Distribution	**Mean ($\bar{x}$)**	**Sample Standard Deviation (s_{n-1})**	**Variance (s^2)**
Fringe	60.567	6.314854582	39.87738839
LBL	62.047	6.604575036	43.62041141
F-L	-1.48	4.737091939	22.44004004

Assume the normal distribution, $X = F - L$. X is calculated by subtracting the move-count for an LBL solve from the move-count for a fringe solve, for every single solve. This created a table of differences ($F - L$), which was analysed. From the mean, it seems that the alternate hypothesis is true, and the fringe solving is faster. However, the spread of the sample must be considered. Since the standard deviation of the population is unknown, a t-distribution needs to be used [Sta20]. The t-value is:

$$t = \frac{\bar{x} - \mu_0}{\frac{s_{n-1}}{\sqrt{n}}} = \frac{-1.48 - 0}{\frac{4.737}{\sqrt{1000}}} = -9.880$$

The p-value: $p = \mathrm{P}(T_{n-1} < t)$. The p-value was calculated using a TI-$n$spire CX. It turns out that the p-value is virtually 0. This is because of the high number of samples and very high degree of freedom for the t-distribution ($df = 999$). Hence, the null hypothesis can be rejected at a 0% significance level, near absolute certainty. The fringe solving technique is faster by an average of 1.48 moves, and the average move-count for solving the 15-puzzle using this technique is roughly 60 moves. However, I also realised that the two methods are far too similar in move-count to make a significant difference. 2 moves can usually be performed in about 150 milliseconds since speed-solvers can do between 12-13 moves per second.

Regardless, statistically speaking, the 15-puzzle is expected to be solved in 60 moves using the most effective technique. At a speed of 12-13 moves per second, this takes 4.5-5 seconds. The world record holder averages this time, which is remarkable, since it would involve executing the most computationally optimal solution every single solve.

4 Conclusion

This investigation was relatively successful in completing the aims set out in the introduction. A proof based analysis of permutations allowed for an understanding of the set of solvable scrambles, addressing the issue of solvability. Once the permutation analysis gave way to an invariant, it was also quite simple to extend the mathematics to other quadrilateral puzzles and n-dimensional puzzles. Unfortunately, the application of group and graph theory was unsuccessful in determining an optimal solution to scrambles. However, a statistical analysis of a sample of 1000 scrambles was sufficient in finding the average move-count to solve the

15-puzzle using a given technique.

Nevertheless, this investigation could lead into more research possibilities. The 15-puzzle and the n-dimensional variants discussed are Euclidean shaped and quadrilateral puzzles. The invariant and permutation analysis was only applied to squares, rectangles, cuboids, etc. However, non-Euclidean shapes and non-quadrilaterals can also be analysed. For example, a triangle, a torus, a dodecahedron, etc. Unordinary shapes can be analysed in future investigations. One might think that this is relatively useless information, however puzzles often lead to interesting mathematical conclusions that can be universally applied. For example, group theory analysis of the game "Set" led to a revolutionary proof in group theory in 2016 [Kla16]. Further investigation into the 15-puzzle can lead to similar revolutionary results.

Personally, the most interesting part of this investigation was the broad application of various mathematics to one puzzle. I used permutations and set theory for an original proof, but was pleasantly surprised to see a group theory and graph theory application even if it didn't lead to tangible results. The drawback of using graph theory only led me to investigate using statistics instead and a large sampling of data. This investigation has taught me that various fields of mathematics can be taken used, applied, and taken advantage of in one problem.

References

[Bas+08] Saugata Basu et al. "Polynomials that Sign Represent Parity and Descartes Rule of Signs". In: *Computational Complexity* 17 (2008), pp. 377–406.

[Cha+] Andrew Chappple et al. "An Analysis of the 15-Puzzle". Louisiana State University.

[Dun08] Corey Dunn. "A proof that Corey finally understands". California State University, Feb. 2008.

[Gra02] Hazel Grant. "The Fifteen Puzzle". In: *Euclidean Geometry Project* (2002). URL: http://www.math.ubc.ca/~cass/courses/m308-02b/projects/grant/fifteen.html.

[Kla16] Erica Klarreich. "Simple Set Game Proof Stuns Mathematicians". In: *Quanta Magazine* (May 2016).

[QHH13] Catherine Quinn, Robert Haese, and Michael Haise. "Mathematics HL (Option): Sets, relations and groups". In: G152 Richmond Road, Marleston, SA 5033, Australia: Haese & Harris Publications, 2013, pp. 68–79.

[Red19] Panda the Red. "Introduction to Group Theory". In: *Cantor's Paradise* (Mar. 2019).

[SS06] Jerry Slocum and Dic Sonneveld. "The 15 Puzzle: How it Drove the World Crazy". In: *The Slocum Puzzle Foundation* (2006).

[Sta20] Kognity Staff. "IBDP Mathematics HL". In: Section 7.6.1. Kognity, 2020.

[SW20] Jerry Slocum and Eric Weisstein. *15 Puzzle*. 2020. URL: https://mathworld.wolfram.com/15Puzzle.html.

[Wei20] Eric Weisstein. "Invariant". In: *Wolfram* (Feb. 2020). URL: https://mathworld.wolfram.com/Invariant.html.