Proceedings of CONF-CDS 2025 Symposium: Data Visualization Methods for Evaluation
DOI: 10.54254/2755-2721/2025.PO25579

Sliding Graph Puzzles: Permutation Groups, Wilson's Theorem, and New Algorithms

Chun Chi Wong[1*], Linxuan Li[2], Ruohan Chen[3]

[1]Shenzhen College of International Education, Shenzhen, China
[2]Beijing New Talent Academy, Beijing, China
[3]Jinqiu International, Qingdao, China
**Corresponding Author. Email: s22367.Wong@stu.scie.com.cn*

Abstract. The 15-puzzle is a classic sliding puzzle consisting of a 4x4 grid with 15 numbered square tiles and one empty space. In 1974, Wilson generalized the 15-puzzle to find the group of permutations on graphs. In this work, we provide a variation of a proof of Wilson's theorem, propose a result for 1-connected and disconnected graphs, find a new manual algorithm for solving sliding graph puzzles, and extend existing computer algorithms on the 15-puzzle to solve any sliding graph puzzle.

Keywords: Sliding puzzle, Permutation group, Graph theory, Manual algorithm

1. Introduction

1.1. History of the 15-puzzle and its generalizations

The well known intellectual game with 15 numbered square counters has a ancient and remarkable history. The game was created by Noyes Chapman and it was first appeared in the 1870s. It spread quickly and soon it became extremely popular in the United States and Europe. This is W. Arens's comments on the influence of 15-puzzle:"Here you could even see the passengers in horse trams with the game in their hands. In offices and shops bosses were horrified by their employees being completely absorbed by the game during office and class hours. Owners of entertainment establishments were quick to latch onto the rage and organized large contests. The game had even made its way into solemn halls of the German Reichstag."In the 1900's, many people worked tirelessly to solve the problems related to the 15-puzzle, but actually some of them can be seen to be unsolvable due to the power of mathematics. Of course, the most famous issue must be the problem proposed by the impish puzzle maker Sam Loyd. He offered a prize of 1000 dollars for the first correct solution to the problem however it has never been claimed.This is because it is impossible to solve this challenge. People first proved it at the end of 19th century by permutation parity. In the solved state, the permutation is even but swapping only 14 and 15 create an odd permutation.Since then, many extensions of the 15-puzzle have been proposed and studied. [1] studied puzzles with more than 1 empty space, [2] studied puzzles with tiles that are rotatable, and puzzles that allowed rotation without an empty space were discussed in [3].A well-known generalization of the 15-puzzle is that of R. M. Wilson in 1974, where numbered tiles are slid across edges in a graph. Formally,

DOI: 10.54254/2755-2721/2025.PO25579

vertices are labeled with numbers, with one vertex labeled the empty space, and we are allowed to swap a numbered label and an adjacent empty label.In his paper [4], he found the group of permutations for non-separable simple graphs. His theorem is sometimes called Wilson's theorem.

1.2. Contents and paper structure

Building on the generalization of the 15-puzzle onto sliding graph puzzles, we find the group of permutations of labels on sliding graph puzzles that are simple and non-separable using similar ideas as [4], by first finding the group on theta graphs and then inducting on the cyclomatic number. We use a more direct approach in the proof, explicitly finding sequences of moves that carries out certain permutations or generates certain groups. This aids the process of following this proof to create a manual algorithm to solving any sliding graph puzzle. Following this manual algorithm, one can theoretically carry out a sequence of moves to solve any scrambled state of any non-separable sliding graph puzzle. The algorithm is recursive, reducing the cyclomatic number of the unsolved portion of the graph until the problem is reduced to solving a theta graph, which is done directly.

We extend Wilson's result on the permutation group for a non-separable graph to one that applies to any finite simple graph, and present a new brief proof of the 1-connected and disconnected cases by discussing the movement of tiles across articulation vertices. We also propose a generalization of the manhattan distance metric to any sliding graph, and explore the performance of the A* and suitably weighted A* algorithms in graphs with different cyclomatic numbers.

In section 2, we present an overview of the group theory and permutation groups needed in later discussions. In section 3, we represent states of the 15-puzzle with a permutation and use group theory to show that only states with even permutations can be achieved. In section 4, we show Wilson's generalization of the 15-puzzle and present our proof of Wilson's theorem. In section 5, we extend Wilson's theorem to 1-connected, disconnected, polygon, and non-simple graphs. In section 6, we present a manual algorithm for solving any non-separable graph puzzle. In section 7, we attempt to generalize A* algorithms on the 15-puzzle to solve any sliding graph puzzle. In section 8, we end the paper with suggestions for future work.

2. A review of group theory and permutation groups

In this section, we introduce group theory and permutation groups to aid the discussion in later sections. We see in section 3 and 4 that states on the 15-puzzle and sliding graph puzzles in general can be represented as permutation, and group theory tools are useful for generating and constructing certain permutation groups, such as that of θ_0 (see section 4.3).

2.1. Groups and relevant theorems

Definition 2.1. A subset S of a group G generates G if every element of G can be expressed as a finite product of elements from S and their inverses.

Definition 2.2. A homomorphism is a function between two groups $\varphi : A \mapsto B$ such that $\varphi(a \cdot b) = \varphi(a) \cdot \varphi(b)$. An isomorphism is a bijective homomorphism. Two groups are isomorphic if there exists an isomorphism from one group to the other.

Definition 2.3. The image of a homomorphism $\mathrm{Im}(\varphi)$ is the subgroup $\{\varphi(a) : a \in A\}$ of B. The kernel of a homomorphism $\ker(\varphi)$ is the subgroup $\{a : \varphi(a) = e_B\}$.

Theorem (Lagrange theorem). For a finite group G and subgroup $H \leq G$, $|H|$ divides $|G|$.

Definition 2.4. A Sylow p-subgroup is a subgroup with order p^k for maximal k.

DOI: 10.54254/2755-2721/2025.PO25579

Theorem (Second Sylow theorem). If P_1 and P_2 are 2 Sylow p-subgroups, then there exists g with $gP_1g^{-1} = P_2$.

Theorem (First isomorphism theorem). Let $f : G \mapsto H$ is a group homomorphism. Then $G/\ker(f) \cong \mathrm{Im}(f)$. In particular, $|G| = |\ker(f)||\mathrm{Im}(f)|$.

2.2. Permutations and permutation groups

Definition 2.5. A permutation of a finite set is a rearrangement of its elements, defined as a bijective function (a one-to-one and onto mapping) from the set to itself. A transposition is a permutation that swaps two elements in a set while leaving all other elements unchanged.

Definition 2.6. A permutation is even if it can be expressed as a product of an even number of transpositions. A permutation is odd if it can be expressed as a product of an odd number of transpositions.

Definition 2.7. The symmetric group S_n is the group of all permutations of a finite set of n elements. The group of all permutations on a set X is denoted by $\mathrm{Sym}(X)$. The alternating group A_n is the group of all even permutations of a finite set of n elements. It is a subgroup of the symmetric group S_n . The alternating group on a set X is denoted by $\mathrm{Alt}(X)$.

Permutations can be expressed using cycle notation, where the elements in the brackets are in one cycle. Each element maps to the next in the cycle, and the last element maps to the first. Every permutation can also be written as a product of transpositions. Below is an example:

$$(1,3,7,2,5,11,10,8,4,9,6,13)(14,15) =$$
$$(1,3)(1,7)(1,2)(1,5)(1,11)(1,10)(1,8)(1,4)(1,9)(1,6)(1,13)(14,15).$$

In this paper, we use the convention that permutations are evaluated from left to right, or in other words, that the group action of S_n on $[n]$ is a right action.

Proposition 2.8. The alternating groups A_n can be generated by 3-cycles.

Proof. By definition, every even permutation can be written as a product of an even number of transpositions. Separating this product into pairs of transpositions, the following three cases arise, where a,b,c,d are distinct elements:

1. $(a, b)(a, b)$ is the identity permutation
2. $(a, b)(a, c) = (a, b, c)$, a 3-cycle
3. $(a, b)(c, d) = (a, b, c)(a, d, c)$, a product of 3-cycles

Since the product of every two transpositions can be expressed as a product of 3-cycles, it follows that the alternating group A_n can be generated by all 3-cycles. □

Proposition 2.9. The alternating group can be generated by (a, b, k) , where a and b are fixed distinct elements, and $k \in [n] - \{a, b\}$.

Proof. Since all alternating group can be generated by 3-cycles, we only need to prove (a, b, k) can generate all 3-cycles.Let a and b be two fixed distinct elements in $[n]$. We show that any 3-cycle (x, y, z) can be expressed as a product of 3-cycles of the form (a, b, k) , where $k \in [n] - \{a, b\}$ .Case 1: $\{x, y, z\} \cap \{a, b\} = \emptyset$ If x, y, z are distinct from a and b, then:

DOI: 10.54254/2755-2721/2025.PO25579

$$(x, y, z) = (a, b, x)(a, b, y)^{-1}(a, b, z).$$

Case 2: If x, y, z share one element with $\{a, b\}$, we consider the following subcases:

Table 1.

Case	Expression
$x = a$	$(a, y, z) = (a, b, y)^{-1}(a, b, z)$
$y = a$	$(x, a, z) = (a, b, z)^{-1}(a, b, x)$
$z = a$	$(x, y, a) = (a, b, x)^{-1}(a, b, y)$
$x = b$	$(b, y, z) = (a, b, y)(a, b, z)^{-1}$
$y = b$	$(x, b, z) = (a, b, z)(a, b, x)^{-1}$
$z = b$	$(x, y, b) = (a, b, x)(a, b, y)^{-1}$

Case3: Two elements equal to a and b
If two elements of {x,y,z} are equal to a and b,we have:

DOI: 10.54254/2755-2721/2025.PO25579

Table 2.

Case	Expression
$\mathrm{x} = \mathrm{a}$ and $\mathrm{y} = \mathrm{b}$	$(a, b, z) = (a, b, z)$
$x = a$ and $z = b$	$(a, y, b) = (a, b, y)^{-1}$
$y = a$ and $z = b$	$(x, a, b) = (a, b, x)$
$x = b$ and $y = a$	$(b, a, z) = (a, b, z)^{-1}$
$x = b$ and $z = a$	$(b, y, a) = (a, b, y)$
$y = b$ and $z = a$	$((x, b, a) = (a, b, x)^{-1}$

In all cases, any 3-cycle (x, y, z) can be expressed as a product of 3-cycles of the form (a, b, k) . Therefore, all 3-cycles can be generated by (a, b, k) . □

3. The permutation group of 15-puzzle

In this section, we represent states of the 15-puzzle as permutations, and demonstrate that the set of all legal permutations in the puzzle is isomorphic to A15 .

According to definition 2.7, the set of all states of the 15-puzzle is a subgroup of Sn . Here is an example:

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

Figure 1. An example state of the 15-puzzle

We can represent this state of the 15-puzzle using cycle notation:

DOI: 10.54254/2755-2721/2025.PO25579

$$\sigma = (1,3,7,2,5,11,10,8,4,9,6,13)(14,15).$$

When we consider that the position of the blank space remains unchanged between the initial and goal states, the number of moves the blank space makes upward should be equal to the number it makes downward, and the number of moves it makes left should be equal to the number it makes right. Therefore, only when the initial state of 15-Puzzle that is an even permutation can be solved. Therefore, the state where tiles 14 and 15 are swapped cannot be solved because it is an odd permutation.

Using this logic, we only prove that the 15-puzzle is a subgroup of the alternating group, and we still need to determine whether all even permutations can be solved. To prove that all even permutations can be solved, we largely follow the process in [5] and we use Propositions 2.8 and 2.9 from the previous section.

Firstly, we construct a 3-cycle $(11,12,15)$. The initial state is shown at the beginning of Fig. 3.2. By sliding these tiles, we generate the cycle $(11,12,15)$.

11	12
15	

⇒

11	12
	15

⇒

	12
11	15

⇒

12	
11	15

⇒

12	15
11	

Figure 2. Process of generating (11,12,15)

The next step is constructing "the long cycle", ρ = (1, 2, 6, 7, 3, 4, 8, 15, 10, 14, 13, 9, 5). This cycle is constructed by temporarily displacing tiles 11 and 12 from their original slots and do a sequence of moves. we first change the position of the right bottom corner to that of Fig. 3.3 and move all the tiles except 11 and 12 through the path shown in Fig. 4, and the 15-puzzle will look like Fig. 5.

15	
12	11

Figure 3. Configuration of the bottom right corner before applying the long cycle

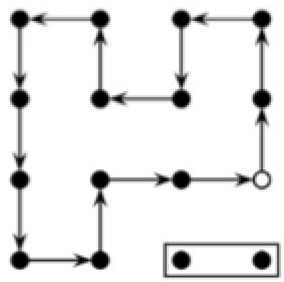

Figure 4. A visual representation of the long cycle

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

Figure 5. State of the 15-puzzle after applying the long cycle once

Finally, we restore the positions of 11 and 12 and this is the entire long cycle ρ .

The purpose of ρ is to move a random tile k to the bottom right corner where the previous three cycle (11,12,15) can act on them. For example, if we want to generate the cycle (3,11,12), we need to apply ρ10 to move the tile 3 to slot 15. Then we can apply the previous 3 cycle $(11,12,15)$, but tile 3 is at slot 15 so the cycle actually swaps tile 11 12 and 3, shown in Fig. 3.6. Finally, we only need to apply ρ−10 to restore all other tiles to their original positions and we can generate the cycle (3,11,12)$, shown in Fig. 3.7.

DOI: 10.54254/2755-2721/2025.PO25579

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

Figure 6. The 15-puzzle after permuting the desired tiles

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

Figure 7. The 15-puzzle after generating(3,11,12)

Through this example, we can see that by conjugating the fixed 3-cycle $(11,12,15)$ with powers of ρ , it can generate all 3-cycles of the form $(11,12,k)$, where $k \in \{1,2,\ldots,15\} - \{11,12\}$. Proposition 2.9 has already proved that 3-cycle (a,b,k) can generate the alternating group A_{15} , so together with the fact that the permutation group of the 15-puzzle is a subgroup of A_{15} , we can conclude that the permutation group of 15-puzzle is isomorphic to A_{15} .

4. Puzzle group of generalized sliding graph puzzles

In this section, we will generalize the permutation group of the 15-puzzle following the discussion in [4], and provide a variation of a proof of Wilson's theorem so as to aid the discussion in section 6.

4.1. Generalizing the 15-puzzle

We introduce the following graph theoretic terms to generalize the 15-puzzle.

Definition 4.1. A graph is a collection of vertices (also called nodes) connected by edges. An edge connects 2 vertices. An undirected graph is a graph where no edge has an associated direction. The set of vertices of a graph G is often denoted as $V(G)$.

Assume from now on that every graph considered is undirected.

Definition 4.2. A simple graph is an undirected graph without loops (edges connecting from a vertex to itself), and any two edges must not have the same end vertices.

Definition 4.3. A path is a sequence of vertices $(x_1, x_2, \ldots, x_n)$ such that any x_i and x_{i+1} are adjacent (connected by an edge). A simple path is a path with a non-repeating sequence of vertices.

Definition 4.4. A connected graph is a graph where there exists a path between any pair of vertices. A disconnected graph is a graph that is not connected. An articulation vertex (also cut vertex) of a connected graph is a vertex that, when removed, yields a disconnected graph. A non-separable (also biconnected, 2-connected) graph is a connected graph with no articulation vertices, or a graph with 1 vertex.

Definition 4.5. A bipartite graph is a graph such that each element can be labeled with 'a' or 'b' with every edge connecting between a vertex labeled 'a' and a vertex labeled 'b'. Equivalently, every path from a vertex to itself must have even length (pass through an even number of edges). A non-bipartite graph is a graph that is not bipartite.

Let G be a finite simple graph and label the vertices with numbers, leaving one vertex blank as the empty space $\varnothing$. This creates a 15-puzzle-like sliding graph puzzle.

DOI: 10.54254/2755-2721/2025.PO25579

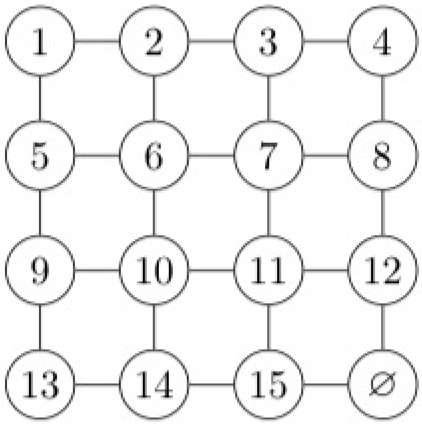

Figure 8. The solved state of the 15-puzzle, represented by a graph with labels

Formally define a labeling on the graph G with $n+1$ vertices as a bijective function $f: V(G) \mapsto \{1,2,...,n,\varnothing\}$ that maps every vertex to its label. Let x be a vertex such that $f(x)=\varnothing$, let two labelings f and g be adjacent if $g(x)=f(y)$ and $g(y)=\varnothing$ for some vertex y adjacent to x (meaning they are connected with an edge), and $f(k)=g(k)$ for all vertices $k \notin \{x,y\}$. Define a move to be the relabeling of a graph by an adjacent labeling. Define the puzzle group of a graph G by P_G, containing a permutation of $\{1,2,...,n\}$ for every labeling f with $s^{-1}(\varnothing)=f^{-1}(\varnothing)$ that can be reached from a sequence of moves starting with the solved state labeling, s, under composition of permutations. This coincides with the homotopy group of the graph. We denote the permutation of labelings e.g. $f(a), f(b), f(c)$ with vertices directly e.g. (abc) .

We first prove the following useful proposition:

Proposition 4.6. Let G be a finite simple connected graph. Then given two solved states s_1 and s_2, the puzzle group with respect to s_1 is isomorphic to the puzzle group with respect to s_2.

Proof. Let $P_G(s_1)$ and $P_G(s_2)$ be the puzzle groups with respect to s_1 and s_2 respectively. We can create an isomorphism $\varphi: P_G(s_1) \mapsto P_G(s_2)$ by starting with the state s_2, moving the blank space to $s_1^{-1}(\varnothing)$, carrying out a permutation p_1 in s_1, and moving the blank space back to $s_2^{-1}(\varnothing)$ along the same path. The permutation p_2 corresponding to this must be in $P_G(s_2)$ since p_1 can be reached with a sequence of moves, meaning p_2 can as well. Since carrying out a different permutation $p_{1'}$ must result in a different permutation $p_{2'}$, φ is injective. If there exists a sequence of moves bringing s_2 to some f_2, corresponding to a permutation in $P_G(s_2)$, then, moving the blank space to $s_1^{-1}(\varnothing)$, there must be a sequence of moves corresponding to a permutation in $P_G(s_1)$, hence φ is surjective. The following 2 processes are equivalent:

1. Moving the blank space to $s_1^{-1}(\varnothing)$
2. Carrying out p_1
3. Moving the blank space to $s_2^{-1}(\varnothing)$
4. Moving the blank space to $s_1^{-1}(\varnothing)$
5. Carrying out q_1
6. Moving the blank space to $s_2^{-1}(\varnothing)$

and

DOI: 10.54254/2755-2721/2025.PO25579

1. Moving the blank space to $s_1^{-1}(\varnothing)$
2. Carrying out p_1q_1
3. Moving the blank space to $s_2^{-1}(\varnothing)$

hence $\varphi(p_1q_1) = \varphi(p_1)\varphi(q_1)$, and so φ is a well-defined homomorphism. Thus, φ is an isomorphism. □

The puzzle group for any non-separable graph can be determined by the following theorem. Wilson's theorem is applied to puzzle groups as follows:

Theorem 4.7 Let G be a finite simple non-separable graph other than a polygon or the graph θ_0 shown in Fig. 4.3. Then, $P_G = \mathrm{sym}\big(V(G) - \{f^{-1}(\varnothing)\}\big)$, unless G is bipartite, in which case $P_G = \mathrm{alt}\big(V(G) - \{f^{-1}(\varnothing)\}\big)$. If $G = \theta_0$, then $P_G \subset \mathrm{sym}\big(V(G) - \{f^{-1}(\varnothing)\}\big)$ is a transitive subgroup of S_6 isomorphic to S_5 .

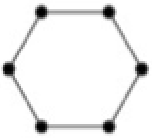

Figure 9. A polygon graph

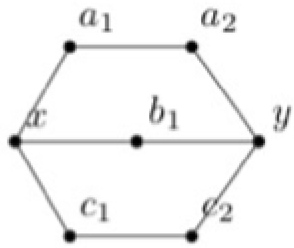

Figure 10. θ_0

Remark. When G is 1-connected (separable and connected), the puzzle group is no longer transitive and this case is dealt with in Theorem 5.1.

4.2. Wilson's theorem on theta graphs

To prove Theorem 4.7, we largely follow Wilson's proof in [2]. It is sufficient to prove a weaker result on theta-graphs, which are obtained by subdivisions of the graph shown in Fig. 4.4 that are simple. Start with some arbitrary theta-graph, and attempt to generate the alternating group or symmetric group from a sequence of moves. Denote a theta graph using the length of the simple paths from x to y. For example, consider the (2,2,3) theta graph in Fig. 4.5, with x , y , a_i , b_i , c_i denoting vertices.

Figure 11. Graph that is subdivided to obtain theta graphs

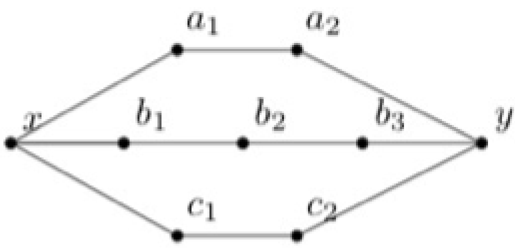

Figure 12. A (2,2,3) theta graph

Since the puzzle group stays unchanged when the solved state is changed, choose a solved state such that $s^{-1}(\varnothing) \notin \{x, y\}$. In the example Fig. 4.5, let $s^{-1}(\varnothing) = c_1$. Consider the following 2 permutations $(a_1a_2yb_3b_2b_1)$ and $(b_3b_2b_1xa_1a_2)$ which are in P_G due to the sequences of moves shown by listing the sequence of vertices the blank label is moved to: $c_1, x, b_1, b_2, b_3, y, a_2, a_1, x, c_1$

DOI: 10.54254/2755-2721/2025.PO25579

and $c_1, c_2, y, a_2, a_1, x, b_1, b_2, b_3, y, c_2, c_1$ respectively. (By moving the blank label along the above sequence, we carry out the permutations in question.) Attempting to generate a group, we prove the following lemma, also proposed in [4]:

Lemma 4.8. Let $S \coloneqq \{x, a_1, \ldots, a_m, y, b_n, \ldots, b_1\}$ and let permutations $p_1 \coloneqq (a_1 a_2 \cdots a_m y b_n \cdots b_2 b_1)$ and $p_2 \coloneqq (b_n \cdots b_2 b_1 x a_1 a_2 \cdots a_m)$. Then:

$$\left\langle p_1, p_2 \right\rangle = \begin{cases} sym(S) & (m+n \text{ odd}, \{m,n\} \neq \{1,2\}) \\ alt(S) & (m+n \text{ even}, \{m,n\} \neq \{2,4\}, m,n \neq 2) \end{cases}$$

Proof. Instead of using the doubly transitive property of S in the proof in[4], we explicitly construct all 3-cycles needed to apply Proposition 2.9, which will aid the discussion in section 6.Let $n \geq m \geq 3$. It is shown in [4] that the permutation

$$p_3 = \left(x b_{m-1} a_1\right) = p_2^{-1} p_1^{m} p_2^{-1} p_1^{1-m} p_2^{-2} p_1 p_2^{-1} p_1 p_2^{2} p_1^{m-1} p_2^{-1} p_1^{2-m} \in \left\langle p_1, p_2 \right\rangle .$$

We can deduce by symmetry and show that

$$p_4 = \left(y b_{n-m+2} a_m\right) = p_1 p_2^{-m} p_1 p_2^{m-1} p_1^{2} p_2^{-1} p_1 p_2^{-1} p_1^{-2} p_2^{1-m} p_1 p_2^{m-2} \in \left\langle p_1, p_2 \right\rangle .$$

Consider the following permutations:

$$p_5 = p_1^{-m} p_3^{-1} p_1^{m} = \left(x y a_1\right)$$
$$p_6 = p_2^{m} p_4^{-1} p_2^{-m} = \left(y x a_m\right)$$
$$p_7 = p_5^{-1} p_6 p_1 p_2^{-1} = \left(x a_1 b_1\right)$$
$$p_8\left(k\right) = p_1^{-k} p_7^{-1} p_1^{k}$$

When k isn't 0 (or a multiple of $m+n+2$), we can generate any 3-cycle $(x a_1 z)$ where z is any element in $A = \{a_2, a_3, \ldots, a_m, y, b_n, \ldots, b_2, b_1\}$ by considering $\prod_{i=1}^{j} p_8\left(i\right) = \left(x a_1 z\right)$ where z is the j^{th} element of A. By Proposition 2.9, $\mathrm{Alt}(S) \leq \langle p_1, p_2 \rangle$ .Letting $m \leq n$ without loss of generality, discuss the excluded cases. When $m = 0, n \neq 0$, $\left(x y b_k\right) = p_2^{k} p_1^{-k}$ so $\mathrm{Alt}(S) \leq \langle p_1, p_2 \rangle$ by Proposition 2.9. When $m = n = 0$, $\mathrm{Alt}(S)$ is the identity, which is trivially a subgroup of $\langle p_1, p_2 \rangle$. When $m = 1, n > 2$, consider $p_3 = p_1^{2} p_2^{-2} p_1^{-1} p_2^{3} p_1^{-2} p_2^{-2} p_1^{2} p_2 p_1^{-3} p_2^{2} = \left(x a_1 y\right)$ and $p_4\left(k\right) = p_1^{-k} p_3 p_1^{k}$. Considering $\prod_{i=0}^{j} p_4\left(i\right) = \left(x a_1 z\right)$ where z is the $(j+1)^{th}$ element of $\{y, b_n, \ldots, b_1\}$ like before, we generate all 3 cycles $(a_1 x z)$ required for Proposition 2.9 to imply that $\mathrm{Alt}(S) \leq \langle p_1, p_2 \rangle$. When $m = n = 1$, consider the 3-cycles $p_1^{-1} p_2^{-1} = \left(y x b_1\right)$ and $p_1 p_2 = (y x a_1)$ and use Proposition 2.9 to show that $\mathrm{Alt}(S) \leq \langle p_1, p_2 \rangle$. When $m = 2, n > 4$, consider $p_3 = p_2^{-1} p_1^{-2} p_2^{-1} p_1^{-1} p_2^{2} p_1 p_2^{2} p_1^{-3} p_2^{-1} p_1^{-1} p_2^{2} p_1 p_2 p_1 = \left(x a_2 a_1\right)$ and $p_4\left(k\right) = p_1^{-k} p_3^{-1} p_1^{k}$ like before,

DOI: 10.54254/2755-2721/2025.PO25579

concluding that $\mathrm{Alt}(S) \leq \langle p_1, p_2 \rangle$. When $m=2, n=3$, consider instead $p_3 = p_2 p_1^{-1} p_2^{-2} p_1^{-1} p_2^{-2} p_1^{-1} p_2^{-1} p_1 = (x a_2 a_1)$ and proceed as before.We have shown that for all cases in the lemma, $\mathrm{Alt}(S) \leq \langle p_1, p_2 \rangle$. When $m+n$ is even, both p_1 and p_2 are even permutations so $\mathrm{Alt}(S) \geq \langle p_1, p_2 \rangle \Rightarrow \mathrm{Alt}(S) = \langle p_1, p_2 \rangle$. When $m+n$ is odd, both p_1 and p_2 are odd permutations so $\mathrm{Alt}(S) \not\geq \langle p_1, p_2 \rangle \Rightarrow \mathrm{Sym}(S) = \langle p_1, p_2 \rangle$. □

The permutation group of any theta graph contains such elements p_1 and p_2. Consider a bipartite theta graph G that is not (2,2,2) or (2,2,4). Any choice of 2 simple paths from x to y must have $m+n$ even, otherwise a subgraph consisting of the 2 paths is a polygon graph of odd length, which meaning G is not bipartite. Verify that we can always choose 2 paths $(x, a_1, \ldots, a_m, y)$ and $(x, b_1, \ldots, b_n, y)$ satisfying $m+n$ even, $\{m,n\} \neq \{2,4\}, (m,n) \neq (2,2)$, and assume that $f^{-1}(\varnothing) = c_i$ for some i by Proposition 4.6. Defining p_1 and p_2 as before, it follows by Lemma 4.8 that P_G must contain the permutation group of $S = \{x, a_1, \ldots, a_m, y, b_n, \ldots, b_1\}$ which is $\mathrm{Alt}(S)$. Since this is an alternating group, it contains the 3-cycles $(a_1 a_2 k)$ for all $k \in S - \{a_1, a_2\}$. We show that $P_G = \mathrm{Alt}\big(V(G) - f^{-1}(\varnothing)\big)$ by generating the 3-cycles $(a_1 a_2 k)$ for all $k \in V(G) - \{a_1, a_2, f^{-1}(\varnothing)\}$:Performing a sequence of moves moving the blank label to $c_i, c_{i+1}, \ldots c_j, y, b_n, \ldots, b_1, x, c_1, \ldots, c_i$ corresponds to a permutation $p_3 = (x b_1 \cdots b_n y c_j \cdots c_{i+1} c_{i-1} \cdots c_1) \in P_G$. Conjugating the 3 -cycle $(a_1 a_2 y)$ by a power of p_3 yields a 3-cycle $(a_1 a_2 k)$ with k in the cycle notation of p_3. Thus, a 3-cycle $(a_1 a_2 k)$ exists in P_G for $k \in V(G) - \{a_1, a_2, f^{-1}(\varnothing)\}$, and we can use Proposition 2.9 giving $P_G \geq \mathrm{Alt}\big(V(G) - f^{-1}(\varnothing)\big)$. We prove the following proposition to show that equality holds:

Proposition 4.9. The puzzle group of a simple finite connected bipartite graph G must be a subgroup of $\mathrm{Alt}\big(V(G) - f^{-1}(\varnothing)\big)$.

Proof. Consider any sequence of moves from the solved state to a labeling f such that $s^{-1}(\varnothing) = f^{-1}(\varnothing)$. There must be an even number of moves since the graph is bipartite, corresponding to a permutation in $\mathrm{Sym}(V(G))$ (containing the empty vertex) which can be written as an even number of transpositions. The resulting permutation must be even and fixes $\varnothing$, so P_G must be a subgroup of $\mathrm{Alt}\big(V(G) - f^{-1}(\varnothing)\big)$. □

So, $P_G = \mathrm{Alt}\big(V(G) - f^{-1}(\varnothing)\big)$ .Now, consider a non bipartite theta graph graph G that is not (1,2,2). We can now choose 2 paths $(x, a_1, \ldots, a_m, y)$ and $(x, b_1, \ldots, b_n, y)$ satisfying $m+n$ odd, $\{m,n\} \neq \{1,2\}$ assuming that $f^{-1}(\varnothing) = c_i$ for some i. P_G contains $\mathrm{Sym}(S)$ by Lemma 4.8, so we can find a transposition $(a_1 k) \in P_G$ for $k \in S$. Conjugating $(a_1 y)$ by a power of $p_3 = (x b_1 \cdots b_n y c_j \cdots c_{i+1} c_{i-1} \cdots c_1)$, we reach every transposition $(a_1 k)$ for k in the cycle notation of p_3, and thus every transposition $(a_1 k)$ for $k \in V(G) - f^{-1}(\varnothing)$ is in P_G. This generates the symmetric group $P_G = \mathrm{Sym}\big(V(G) - f^{-1}(\varnothing)\big)$.

4.3. Considerations for exceptional theta graphs

For the special cases excluded from the above discussion, we discuss each of them individually.

DOI: 10.54254/2755-2721/2025.PO25579

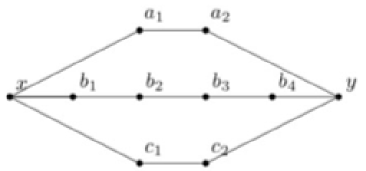

Figure 13.A (2,2,2) theta graph

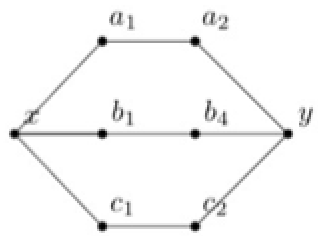

Figure 14.A (2,2,4) theta graph

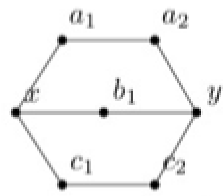

Figure 15. A (1,2,2) theta graph

Let G be the (2,2,2) theta graph shown in Fig. 4.6 and let $f(b_2)=\varnothing$. Define the permutations $p_1=(ya_2a_1xb_1)$, $p_2=(yc_2c_1xb_1)$, and $p_3=(a_2a_1xc_1c_2)$, noting that they are all in P_G . [4] considered a 3-cycle generated from similar permutations, but we consider an easier way to generate a 3-cycle $p_4=p_1^{-2}p_3p_1^2p_2^{-1}=\left(yc_2c_1\right)\in P_G$. We can now conjugate p_4 by powers of p_1 and use Proposition 2.9 and the fact that G is bipartite to conclude that $P_G=\mathrm{Alt}\left(V\left(G\right)-f^{-1}\left(\varnothing\right)\right)$ .Let G be the (2,2,4) theta graph shown in Fig. 14 and let $f(b_4)=\varnothing$. Define $p_1=(ya_2a_1xb_1b_2b_3)$, $p_2=(yc_2c_1xb_1b_2b_3)$, and $p_3=(a_2a_1xc_1c_2)$ in P_G . Consider the 3-cycle $p_4=p_1^{-3}p_3^{-1}p_2p_1^{-2}p_2^2=\left(yc_2c_1\right)$. Conjugating by a power of p_1 , we conclude that $P_G=\mathrm{Alt}\left(V\left(G\right)-f^{-1}\left(\varnothing\right)\right)$ .Let G be the (1,2,2) theta graph shown in Fig. 15 and let $f(b_1)=\varnothing$. Define $p_1=(ya_2a_1x)$, $p_2=(yc_2c_1x)$, $p_3=(a_2a_1xc_1c_2)$, $p_4=(c_1c_2ya_2a_1)$, and claim that $\langle p_1,p_2,p_3,p_4\rangle$ contains all permutations corresponding to a sequence of moves fixing the position of the blank label. Consider an arbitrary sequence of moves represented by a sequence of positions of the blank label. Assume that this sequence never 'backtracks on itself', i.e. there is no consecutive a,b,a in the sequence for 2 vertices a,b . This is reasonable since canceling the b,a has no effect on the permutation of labels. Split this sequence of positions for every instance of b_1 not at the beginning or end. Each subsequence must correspond to p_1 or its inverse, p_2 or its inverse, a power of p_3 , or a power of p_4 . So the above claim is true.Now, consider the permutation $p_5=p_2^{-1}p_1p_2^{-1}p_1^2=\left(a_2c_2\right)\left(a_1x\right)\left(c_1y\right)$. We claim that this, along with p_3 generate a transitive subgroup of S_6 isomorphic to S_5 with the following proposition:

Proposition 4.10. The subgroup $H=\langle(15)(23)(46),(12345)\rangle<S_6$ is isomorphic to S_5 .

[4]constructed this group by considering the action of the group of linear fractional transformations $\mathrm{PGL}_2(\mathbb{F}_5)$ on the set $\mathbb{F}_5\cup\infty$. We follow the construction in [6] instead:

Proof. Consider all the subgroups of S_5 with order 5:

$$P_1=\langle(12345)\rangle$$
$$P_2=\langle(12354)\rangle$$
$$P_3=\langle(12453)\rangle$$
$$P_4=\langle(12534)\rangle$$
$$P_5=\langle(12435)\rangle$$
$$P_6=\langle(12543)\rangle$$

These are all the Sylow 5-subgroups of S_5 , since no subgroups of order 25 exist by Lagrange's theorem. By the Second Sylow theorem, there exists an element $g\in S_5$ with $g^{-1}P_ig=P_j$ for any

DOI: 10.54254/2755-2721/2025.PO25579

i, j , meaning S_5 can act transitively on the set X of Sylow 5-subgroups by conjugation. Consider the homomorphism from this group action $f: S_5 \mapsto S_6$. The kernel of f must be normal, so it is isomorphic to S_5 , A_5 , or the trivial subgroup. $\mathrm{Im}(f)$ must be a transitive subgroup of S_6 , so it has order at least 6. By the first isomorphism theorem, $|S_5| = |\ker(f)||\mathrm{Im}(f)|$, so it follows that $|\ker(f)| \leq 20 \Rightarrow \ker(f)$ is trivial . So the isomorphism from $S_5/\ker(f)$ to $\mathrm{Im}(f)$ is an isomorphism between S_5 and $\mathrm{Im}(f)$ .We show now that $\mathrm{Im}(f) = H$. The generator $(12345) \in S_6$ corresponds to an element $g = (12543) \in S_5$ satisfying $g^{-1}P_1g = P_2, g^{-1}P_2g = P_3, g^{-1}P_3g = P_4, g^{-1}P_4g = P_5, g^{-1}P_5g = P_1, g^{-1}P_6g = P_6$, so $(12345) \in \mathrm{Im}(f)$. The generator $(15)(23)(46) \in S_6$ corresponds to an element $g = (34) \in S_5$ satisfying $g^{-1}P_1g = P_5, g^{-1}P_2g = P_3, g^{-1}P_3g = P_2, g^{-1}P_4g = P_6, g^{-1}P_5g = P_1, g^{-1}P_6g = P_4$, so $(15)(23)(46) \in \mathrm{Im}(f)$. So, $H \leq \mathrm{Im}(f)$. But the elements (12543) and (34) generate S_5 , so the elements (12345) and $(15)(23)(46)$ generate $\mathrm{Im}(f)$, meaning $H \geq \mathrm{Im}(f) \Rightarrow H = \mathrm{Im}(f)$. □

We show that $p_1, p_2, p_4 \in \langle p_3, p_5 \rangle$:

$$\begin{aligned} p_2 &= p_5p_3p_5p_3^2p_5p_3^3 && \in \langle p_3, p_5 \rangle \\ p_1 &= p_2^2p_5^{-1}p_2^{-2} && \in \langle p_3, p_5 \rangle \\ p_4 &= p_2p_5p_2^{-1} && \in \langle p_3, p_5 \rangle \end{aligned}$$

So, $P_G = \langle p_3, p_5 \rangle \cong S_5$. Since this group is transitive, G is a transitive subgroup of S_6 isomorphic to S_5 .

4.4. Inducting on the cyclomatic number

We complete the proof of Theorem 4.7 by inducting on the cyclomatic number $|E(G)| - |V(G)| + 1$, discussing bipartite and non-bipartite graphs separately. We prove the following theorem, first introduced in [7], to help with induction:

Theorem 4.11. Let G be a simple finite non-separable graph and denote its cyclomatic number by $\beta(G) \geq 2$. Suppose that H is a non-separable proper subgraph of G with cyclomatic number $\beta(G) - 1$. Then we can write $G = H \cup A$, where A is a subgraph of G with cyclomatic number 0 and $H \cap A$ consists only of the ends of A.

Proof. Take any subgraph H satisfying the properties in the theorem, and let n be the number of vertices in G but not in H . Since H has cyclomatic number $\beta(G) - 1$, the number of edges in G but not in H must be $n + 1$. Since G is non-separable, the edges and vertices not in H must connect to at least 2 vertices in H . Let A include the edges and vertices not in H , and the vertices in H connected to an edge not in H . Since A must have cyclomatic number 0, it must be possible to obtain A from subdivisions of the graph below:

Figure 16. Graph which is subdivided to obtain A

Since G is non-separable, the ends of this graph must be in H . If any other vertices in A are in H , the cyclomatic number of G increases by 1, making H have cyclomatic number $\beta(G) - 2$

DOI: 10.54254/2755-2721/2025.PO25579

. So, we can write $G = H \cup A$ where $H \cap A$ consists only of the ends of A. □

We now begin the induction, constructing explicitly the permutations required to generate the puzzle group, without relying on the doubly transitive property of P_G used in [4] and [8].The bipartite case: Having proven that Theorem 4.7 is true for bipartite non-separable graphs with cyclomatic number 2 (which are the theta graphs), we assume that Theorem 4.7 is true for bipartite non-separable graphs with cyclomatic number $k \geq 2$. Take an arbitrary bipartite non-separable graph of cyclomatic number $k+1$ and write $G = H \cup A$ by Theorem 4.11, creating a bipartite graph H of cyclomatic number k whose puzzle group is $P_H = \mathrm{Alt}\left(V(H) - f^{-1}(\varnothing)\right)$. Since this graph is bipartite, by Proposition 4.9, P_H is a subgroup of $\mathrm{Alt}\left(V(H) - f^{-1}(\varnothing)\right)$ and P_G does not contain $\mathrm{Sym}\left(V(H) - f^{-1}(\varnothing)\right)$. Letting x and y be the ends of A , let the simple path from x to y in A pass through vertices $k_1, k_2, \ldots, k_m$, and find a simple path from y to x in H that do not pass through some a_1, a_2 in H , letting it pass through vertices $v_1, v_2, \ldots, v_n$ (One can verify that this is always possible). Assuming $f(\varnothing) = y$ by Proposition 4.6, we can move the blank space to vertices $k_m, \ldots, k_1, x, v_n, \ldots, v_1, y$ to create a permutation $p = (v_1 v_2 \cdots v_n x k_1 k_2 \cdots k_n) \in P_G$. Conjugating the 3-cycle $(a_1 a_2 x)$ by a power of p can create all 3-cycles $(a_1 a_2 k)$ for $k \in A$. Together with the fact that $P_H = \mathrm{Alt}\left(V(H) - f^{-1}(\varnothing)\right)$ and by Proposition 2.9 and Proposition 4.9, $P_G = \mathrm{Alt}\left(V(G) - f^{-1}(\varnothing)\right)$ .The non-bipartite case: Ensure first that given a graph with cyclomatic number 3 containing a subgraph θ_0 , that we can always write $G = H \cup A$ where H a non-bipartite graph that is not θ_0 . Consider all such possible types of graphs, highlighting in red components in G that are not in H, and highlighting in blue the components (possibly and sometimes necessarily subdivided) that could have been removed to yield

H=θ0. The below consideration is omitted in [4] and partially omitted in [8]:

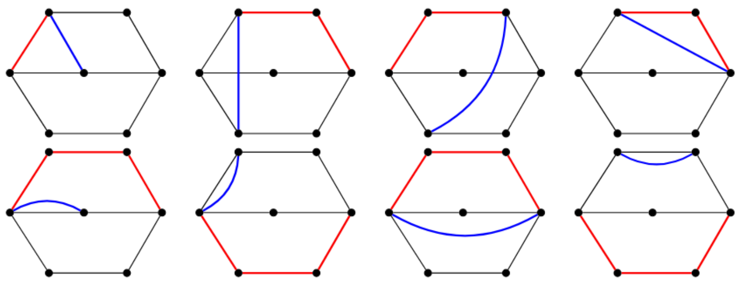

Figure 17. All possible cases whereθ_0 could be chosen as the subgraph H

Assume that Theorem 4.7 is true for non-bipartite non-separable graphs with cyclomatic number $k \geq 2$. Take an arbitrary non-bipartite non-separable graph G with cyclomatic number $k+1$ and we can always write $G = H \cup A$ where H is a non-bipartite graph of cyclomatic number k whose puzzle group is $P_H = \mathrm{Sym}\left(V(H) - f^{-1}(\varnothing)\right)$ (Since $H \neq \theta_0$). Letting x and y be the ends of A , let the path from x to y in A pass through vertices $k_1, k_2, \ldots, k_m$, and find a path from y to x in H that do not pass through some $a \in H$, letting it pass through vertices $v_1, v_2, \ldots, v_n$ (One can verify that this is always possible). As before, we can create a permutation $p = (v_1 v_2 \cdots v_n x k_1 k_2 \cdots k_n) \in P_G$. Conjugating the transposition (ax) by a power of p can create all transpositions (ak) for $k \in A$ which, together with the fact that $P_H = \mathrm{Sym}\left(V(H) - f^{-1}(\varnothing)\right)$, generate the symmetric group $P_G = \mathrm{Sym}\left(V(G) - f^{-1}(\varnothing)\right)$ .Theorem 4.7 is now proved.

DOI: 10.54254/2755-2721/2025.PO25579

5. Extensions of Wilson's theorem

In this section, we extend Theorem 4.7 to 1-connected, disconnected graphs, and polygon graphs. This consideration has already been made by [7] on another theorem in [2], and [10] considered the one-connected case without proof. We give a new brief proof of Theorem 5.1 in this section, an extension of Theorem 4.7:

Theorem 5.1. Let G be a finite simple graph. If $G \neq \theta_0$ is a non-separable graph that is not a polygon, $P_G = \mathrm{sym}\left(V(G) - \{f^{-1}(\varnothing)\}\right)$, unless G is bipartite, in which case $P_G = \mathrm{alt}\left(V(G) - \{f^{-1}(\varnothing)\}\right)$. If $G = \theta_0$, then $P_G \subset \mathrm{sym}\left(V(G) - \{f^{-1}(\varnothing)\}\right)$ is a transitive subgroup of S_6 isomorphic to S_5. If G is a polygon, $P_G \subset \mathrm{sym}\left(V(G) - \{f^{-1}(\varnothing)\}\right)$ is isomorphic to the cyclic group $\mathbb{Z}_{n-1}$. If G is 1-connected (separable and connected) with non-separable components H_i, P_G is the direct product $\prod_i P_{G_i}$, where G_i is a non-separable component of G. If G is disconnected, P_G is the puzzle group of the connected component containing $f^{-1}(\varnothing)$.

Proof. If G is non-separable and is not a polygon, this is Theorem 4.7. Let G be a polygon of length n with vertices labeled a_1 to a_n and $f(a_n) = \varnothing$. Then, the only paths from a_n to a_n without 'backtracing on itself' is $a_{n-1}, \ldots, a_2, a_1, a_n$ and its reverse, corresponding to $p = (a_1 a_2 \cdots a_{n-1})$ and its inverse. It follows that $P_G = \langle p \rangle \cong \mathbb{Z}_{n-1}$.Let G be a 1-connected graph with non-separable components $G_1, G_2, \ldots, G_n$ and let $f^{-1}(\varnothing) \in G_1$. Since G is connected, there exists a sequence of moves bringing the blank label to some vertex $v \in G_i$ such that no other vertex in this sequence belongs in G_i. Then, we can carry out a sequence of moves corresponding to permutations in P_{G_i}, and move the blank label back to $f^{-1}(\varnothing)$ using the same path. This implies that $P_{G_i} < P_G$. Since the graph is 1-connected, the blank space cannot pass through any articulation vertex twice in the same direction as there does not exist a path between the 2 components not passing through the articulation vertex, so the permutations in G_i and any other component must be disjoint. So with $h \in G_i$ and $g \in G$, $ghg^{-1} \in G_i$, hence G_i is a normal subgroup. So, G is the direct product of all non-separable components G_i.If G is disconnected, then the labels of the connected components of G that do not contain $f^{-1}(\varnothing)$ cannot move, and so P_G is the puzzle group of the connected component of G. □

We can also extend further and consider puzzle groups of undirected, non-simple graphs. However, loops and extra edges do not affect the puzzle group of the graph, so the puzzle group of some finite undirected graph is the puzzle group of the simple graph obtained from removing loops and extra edges. This has also been considered in [2].

6. A manual algorithmic solution to sliding graph puzzles

In this section, we provide a method to solve any sliding graph puzzle with a non-separable graph, using ideas from the proof of Theorem 4.7, and adopting a recursive approach.The objective is to algorithmically carry out a sequence of moves that returns a scrambled state f to a solved state s. Considering non-separable graphs only, we can find a path from $f^{-1}(\varnothing)$ to $s^{-1}(\varnothing)$. Moving the blank label along this path, we reach a scrambled state g such that $g^{-1}(\varnothing) = s^{-1}(\varnothing)$. We then find a sequence of moves bringing g to s. First, we show that for both non-bipartite graphs and bipartite graphs, we are able to perform recursion until the graph we need to solve is a theta graph.

DOI: 10.54254/2755-2721/2025.PO25579

6.1. The non-bipartite case

Assume that G is non-separable, non-bipartite, not a polygon, and has cyclomatic number $\beta(G) \geq 2$. By Theorem 4.11, we can continuously write $G = H \cup A$ such that $H \neq \theta_0$ is non-bipartite. We 'solve' A by performing transpositions until $g(v) = s(v)$ for all $v \in A - \{x, y\}$, where x, y are the 2 ends of A . To perform such a transposition on labels $g(a), g(b)$ for $a \in A - \{x, y\}$, $b \in H$, we:

1. Find a cyclic path X passing x and y and not passing through some $c \in H$ as in the proof of Theorem 4.7 above.
2. Transpose $g(b)$ and $g(c)$ by recursion since $b, c \in H$
3. Move the blank label to y
4. Conjugate the transposition (cx) inside H by p , a power of the permutation resulting from X moving a to x
5. Move the blank label back to $g^{-1}(\varnothing)$
6. Transpose $g(b), g(c)$ back

After A is 'solved', we can solve the graph H recursively, moving $\varnothing$ into H and back by Proposition 4.6 if $s^{-1}(\varnothing) \in A - \{x, y\}$. Note that if the permutation we need is on labels $f(a), f(b)$ with $a, b \in A - \{x, y\}$, we can find $d \in H$ and carry out $(ad)(bd)$ .Below is an example of solving a non-bipartite graph puzzle using the above algorithm:

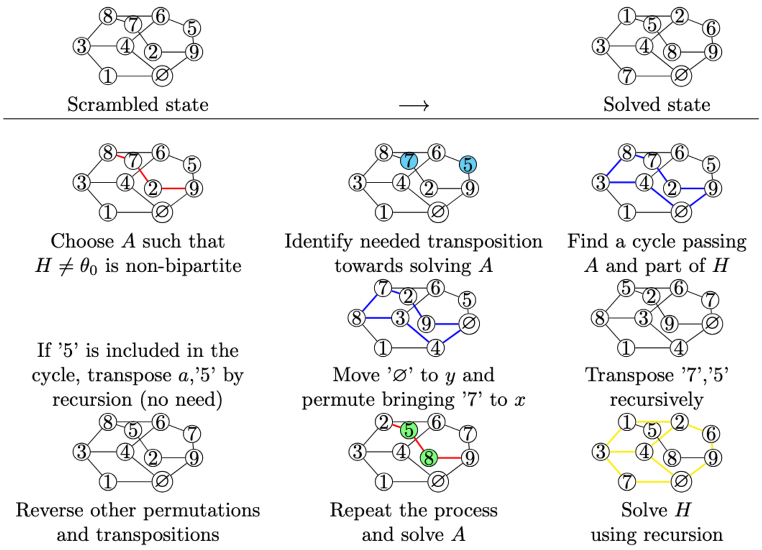

Figure 18. An example of solving an arbitrary sliding graph puzzle recursively

6.2. The bipartite case

For a bipartite, non-separable graph G that is not a polygon and has cyclomatic number $\beta(G) \geq 2$, we write $G = H \cup A$ and solve A by performing 3-cycles until $g(v) = s(v)$ for all $v \in A - \{x, y\}$, where x, y are the 2 ends of A . To perform such a 3-cycle on labels $g(a), g(b), g(e)$ for $a \in A - \{x, y\}$, $b, e \in H$:

1. Find a cyclic path X passing x and y and not passing through some $c, d \in H$ (not necessarily distinct from b, e) as in the proof of Theorem 4.7

DOI: 10.54254/2755-2721/2025.PO25579

• If we could find $b = c$ and $d = e$, there is no need to do any permutation
• If $b = c$ but $d \neq e$ (or vice versa), find a distinct vertex $f \in H$ and carry out (dfe)
• If $b \neq c$ and $d \neq e$, carry out $(bc)(de) = (bcd)(bed)$

2. Move the blank label to y

3. Conjugate the 3-cycle (xcd) by p , a power of the permutation resulting from X , moving a to x

4. Revert all other permutations and moves carried out

After A is solved, we solve the graph H recursively. Note that if the permutation we need is on labels $a, b \in A - \{x, y\}$ and $e \in H$, find a distinct $c \in H$ and carry out $(afe)(bef)$. If $a, b, e \in A - \{x, y\}$, then find distinct $f, g \in H$ and carry out $(afg)(bgf)(cfg)(agf)$.

6.3. Solving theta graphs

We need to generate any transposition on a non-bipartite theta graph $\neq \theta_0$ and any 3-cycle on a bipartite theta graph to successfully carry out the algorithm. For the below discussion, assume that $\varnothing$ has been moved to b_i by the proof of Proposition 4.6, and will be moved back after the permutation is applied.For a bipartite theta graph, we can follow the proof for Proposition 2.9 and express the 3-cycle as a product of 2 3-cycles in the form $(a_1 a_2 k)$, and then follow the proof of Theorem 4.7 to express each $(a_1 a_2 k)$ as $(a_1 a_2 y)$ conjugated by an appropriate power of $p_3 = (x c_1 \cdots c_m y b_j \cdots b_{i+1} b_{i-1} \cdots b_1)$ if $k = c_i$ for some i . In any case, follow the proof of Lemma 4.8 to generate $(a_1 a_2 z) = (x a_1 z)^{-1} (x a_1 a_2)$ for $z \neq c_i$ for any i by a product of permutations $p_8(i)$ as defined in the proof, expressing the original desired 3 cycle as a combination of p_1 and p_2 and their inverses. We can then move the empty space according to the proof of Theorem 4.7 to carry out permutations p_1 and p_2 .For a non-bipartite theta graph $\neq \theta_0$, to perform any transposition (qr) we choose 2 paths from x to y with $m + n$ odd as in the proof of Theorem 4.7, and express $(qr) = (st)(rst)(qrs)$ such that s, t are in the 2 paths. Carry out (st) by performing p_1 as defined in Lemma 4.8, and returning all elements other than s, t by carrying out 3-cycles in the same way as the above procedure for bipartite graphs. Since the resulting permutation is odd, it must be (st) .To solve a theta graph $\neq \theta_0$, we can carry out 3-cycles and a transposition if needed using the above process to return the graph to the solved state. To solve $G = \theta_0$, following the proof of Proposition 4.10, map the permutation p of labelings bringing f to the solved state s onto an element $g \in S_5$ by considering what conjugate $g^{-1} P_i g$ is equal to. Express g as a product of transpositions and express this as a product of transpositions adjacent in the cycle (12543) . Express each transposition as (34) conjugated by a power of (12543) , and map this back to the puzzle group to reach an expression of p in terms of p_3 and p_5 . Following the proof of Proposition 4.10, express instances of p_5 as $p_2^{-1} p_1 p_2^{-1} p_1^2$, and we can perform the sequence of moves corresponding to this permutation to solve the puzzle.

7. Generalizations on existing algorithms to sliding graph puzzles

In this section, we attempt to generalize computer algorithms to solve any sliding graph puzzle. The 15-puzzle is a popular test bed for path finding algorithms, and many efficient and optimized algorithms for solving the 15-puzzle have emerged over the years. Usually, such solvers use the A* algorithm or Iterative deepening A* (IDA*) algorithms, with common heuristic function for best first search algorithms being manhattan distance, linear conflict, and walking distance. However,

DOI: 10.54254/2755-2721/2025.PO25579

many techniques being used in 15-puzzle solvers are difficult to generalize.Letting f and s denote the scrambled and solved states respectively, we can implement the manhattan distance heuristic on any sliding graph puzzle by finding the shortest distance from $f^{-1}(n)$ to $s^{-1}(n)$ and summing across all labels n excluding $\varnothing$. However, as the complexity of the puzzle (which can be described by the cyclomatic number) decreases and the number of nodes increase, the advantage of best-first search over breadth-first search decreases. Table 7.1 shows the average of the number of expanded nodes when finding a solution over 500 random scrambled states.

Table 3. A comparison between A* and BFS algorithms for three graphs

Graph	Cyclomatic number	Expanded nodes (A* w/ manhattan	Expanded nodes (BFS)	Percentage Decrease
(2,2,3) theta graph	2	48152.3	185222	74.003%
8-puzzle	4	2017.29	84869.2	97.623%
8-puzzle, diagonal movements allowed	12	1492.69	162282	99.080%

Given the additional time required in computing the manhattan distance metric for each expanded node, the A* algorithm using the manhattan distance heuristics can sometimes be worse than the BFS algorithm time-wise. A possible reason why the manhattan distance heuristic performs so badly for low complexity graphs is because it severely underestimates the remaining number of moves, and the relative error of the heuristic $\epsilon = \frac{h^* - h}{h^*}$ tends to 1. The effective branching factor b^ϵ approaches the branching factor b , and the time complexity approaches $O(b^d)$, that of BFS (see [10]).Sacrificing the optimality of solution, we can improve efficiency by using a rough estimate of the remaining moves. We multiply the manhattan distance heuristic by an appropriate constant W , which we calculated by averaging the ratio between the manhattan distance heuristic and the number of actual moves required across a random sample of scrambled states. This weighted A* turns the algorithm pessimistic, but as shown by the table below, solutions remain near-optimal solutions while runtime decreases:

Table 4. A comparison between A* and Weighted A* algorithms for three graphs

Graph	Expanded nodes (A*)	Expanded nodes (WA*)	% decrease (nodes)	% increase (solution moves)	Weight W
(2,2,3) theta graph	48152.3	27540.3	42.806%	16.82%	4.0
Fig. 6.1	31332.7	9691.50	69.069%	6.96%	2.0
3×4 puzzle, diagonal movements allowed	67138.1	3336.26	95.031%	4.80%	1.4

8. Conclusion and further prospects

We end this paper with several suggestions for future work.In section 5, we presented some straightforward extensions of Wilson's theorem, deducing the group of permutations for 1-connected, disconnected, cyclic, and non-simple graphs. A possible point of interest is to extend further and find this puzzle group for directed graphs.In section 6, a manual algorithm was developed, capable of solving any sliding graph puzzle with a non-separable graph. It could be interesting to develop similar manual algorithms for other generalizations of the 15-puzzle and

DOI: 10.54254/2755-2721/2025.PO25579

sliding graph puzzles, such as the sliding graph puzzle with where tiles can rotate. Another possible focus is on a specific family of graphs, such as theta graphs.The algorithm presented in section 6 has numerous inefficiencies and has considerable room for improvement. Due to the heavy reliance on performing 3-cycles and transpositions, which often take hundreds of moves on graphs with a low cyclomatic number, solutions to scrambled states developed using this algorithm are often unnecessarily long. A topic of interest is to develop an efficient algorithm that can solve any sliding graph puzzle, while staying human-solvable.The heuristic used in the A* algorithm implementation in section 7 can possibly be improved upon. It is evident that using only manhattan distance as a heuristic leaves the A* search algorithm lost, especially on graphs with lower complexity. Furthermore, with a heuristic that does not correspond well with the actual number of remaining moves, even the weighted algorithm has a tendency to waste moves. This was especially observed for low-complexity graphs such as the (2,2,3) theta graph. We wonder if there exists a more suitable heuristic for general sliding graph puzzles.

References

[1] F. Brunck and M. Kwan, “Books, Hallways and social butterflies: A note on sliding block puzzles, " Math Intelligencer 47, 2025, pp. 52–65.
[2] P. Garcia, A. Hanson, D. Jensen, and N. Owen, “Sliding block puzzles with a twist: On Segerman’s 15 + 4 puzzle”, arXiv preprint, arXiv: 2210.16930, 2022
[3] C. Yang, “Sliding puzzles and rotating puzzles on graphs, ” Discrete Mathematics, vol. 311, no. 14, pp. 1290–1294, 2011.
[4] R. M. Wilson, “Graph puzzles, homotopy, and the alternating group, " Journal of Combinatorial Theory, Series B, vol. 16, no. 1, 1974, pp. 86-96.
[5] Robert A. Beeler, “The Fifteen Puzzle and Related Puzzles (Supplemental Material for Intro to Modern Algebra), " June 28, 2022
[6] J. Gerald and J. Rotman, “Outer Automorphisms of S6, ” The American Mathematical Monthly, vol. 89, no. 6, 1982, pp. 407–410.
[7] H. Whitney, “Non-Separable and Planar Graphs, ” Transactions of the American Mathematical Society, vol. 34, no. 2, 1932, pp. 339-362.
[8] T. Howe, “Two Approaches to Analyzing the Permutations of the 15 Puzzle, ” 2017.
[9] A. F. Archer, “A Modern Treatment of the 15 Puzzle, ” The American Mathematical Monthly, vol. 106, pp. 793-799, 1999.
[10] S. J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd ed., Prentice Hall, 2010, pp. 98-103.