Cyclic Shift Problems on Graphs*

Kwon Kham Sai, Ryuhei Uehara, and Giovanni Viglietta

School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Japan. \{saikwonkham,uehara,johnny\}@jaist.ac.jp

Abstract. We study a new reconfiguration problem inspired by classic mechanical puzzles: a colored token is placed on each vertex of a given graph; we are also given a set of distinguished cycles on the graph. We are tasked with rearranging the tokens from a given initial configuration to a final one by using cyclic shift operations along the distinguished cycles. We first investigate a large class of graphs, which generalizes several classic puzzles, and we give a characterization of which final configurations can be reached from a given initial configuration. Our proofs are constructive, and yield efficient methods for shifting tokens to reach the desired configurations. On the other hand, when the goal is to find a shortest sequence of shifting operations, we show that the problem is NP-hard, even for puzzles with tokens of only two different colors.

Keywords: cyclic shift puzzle, permutation group, NP-hard problem

1 Introduction

Recently, variations of reconfiguration problems have been attracting much interest, and several of them are being studied as important fundamental problems in theoretical computer science [8]. Also, many real puzzles which can be modeled as reconfiguration problems have been invented and proposed by the puzzle community, such as the 15-puzzle and Rubik’s cube. Among these, we focus on a popular type of puzzle based on cyclic shift operations: see Fig. 1. In these puzzles, we can shift some elements along predefined cycles as a basic operation, and the goal is to rearrange the pieces into a desired pattern.

In terms of reconfiguration problems, this puzzle can be modeled as follows. The input of the problem is a graph \( G = (V,E) \), a set of colors \( \text{COL} = \{1,2,\ldots,c\} \), and one colored token on each vertex in \( V \). We are also given a set \( C \) of cycles of \( G \). The basic operation on \( G \) is called “shift” along a cycle \( C \) in \( C \), and it moves each token located on a vertex in \( C \) into the next vertex along \( C \). This operation generalizes the token swapping problem, which was introduced by Yamanaka et al. [11], and has been well investigated recently. Indeed, when we restrict each cycle in \( C \) to have length two (each cycle would correspond to an edge in \( E \)), the cyclic shift problem is equivalent to the token swapping problem.

* This work is partially supported by KAKENHI grant numbers 17H06287 and 18H04091.
In the mathematical literature, the study of permutation groups and their generators has a long history. An important theorem by Babai [1] states that the probability that two random permutations of \( n \) objects generate either the symmetric group \( S_n \) (i.e., the group of all permutations) or the alternating group \( A_n \) (i.e., the group of all even permutations) is \( 1 - \frac{1}{n} + O(n^2) \). However, the theorem says nothing about the special case where the generators are cycles.

In [4], Heath et al. give a characterization of the permutations that, together with a cycle of length \( n \), generate either \( A_n \) or \( S_n \), as opposed to a smaller permutation group. On the other hand, in [7], Jones shows that \( A_n \) and \( S_n \) are the only finite primitive permutation groups containing a cycle of length \( n - 3 \) or less. However, his proof is non-constructive, as it heavily relies on the classification of finite simple groups (and, as the author remarks, a self-contained proof is unlikely to exist). In particular, no non-trivial upper bound is known on the distance of two given permutations in terms of a set of generators.

The computational complexity of related problems has been studied, too. It is well known that, given a set of generators, the size of the permutation group they generate is computable in polynomial time. Also, the inclusion of a given permutation \( \pi \) in the group is decidable in polynomial time, and an expression for \( \pi \) in terms of the generators is also computable in polynomial time [2].

In contrast, Jerrum showed that computing the distance between two given permutations in terms of two generators is PSPACE-complete [6]. However, the generators used for the reduction are far from being cycles.

In this paper, after giving some definitions (Section 2), we study the configuration space of a large class of cyclic shift problems which generalize the puzzles in Fig. 1 (Section 3). We show that, except for one special case, the permutation group generated by a given set of cycles is \( S_n \) if at least one of the cycles has even length, and it is \( A_n \) otherwise. This result is in agreement with Babai’s theorem [1], and shows a similarity with the configuration space of the (generalized) 15-puzzle [10]. Moreover, our proofs in Section 3 are constructive, and yield polynomial upper bounds on the number of shift operations required to reach a given configuration. This is contrasted with Section 4, where we show that finding a shortest sequence of shift operations to obtain a desired configuration is NP-hard, even for puzzles with tokens of only two different colors.
2 Preliminaries

Let $G = (V, E)$ be a finite, simple, undirected graph, where $V$ is the vertex set, with $n = |V|$, and $E$ is the edge set. Let $\text{Col} = \{1, 2, \ldots, c\}$ be a set of colors, where $c$ is a constant. A token placement for $G$ is a function $f : V \rightarrow \text{Col}$: that is, $f(v)$ represents the color of the token placed on the vertex $v$. Without loss of generality, we assume $f$ to be surjective.

Let us fix a set $C$ of cycles in $G$ (note that $C$ does not necessarily contain all cycles of $G$). Two distinct token placements $f$ and $f'$ of $G$ are adjacent with respect to $C$ if the following two conditions hold: (1) there exists a cycle $C = (v_1, v_2, \ldots, v_j)$ in $C$ such that $f'(v_{i+1}) = f(v_i)$ and $f'(v_1) = f(v_j)$ or $f'(v_i) = f(v_{i+1})$ and $f'(v_j) = f(v_1)$ for $1 \leq i \leq j$, and (2) $f'(w) = f(w)$ for all vertices $w \in V \setminus \{v_1, \ldots, v_j\}$. In this case, we say that $f'$ is obtained from $f$ by shifting the tokens along the cycle $C$. If an edge $e \in E$ is not spanned by any cycle in $C$, $e$ plays no role in shifting tokens. Therefore, without loss of generality, we assume that every edge is spanned by at least one cycle in $C$.

We say that two token placements $f_1$ and $f_2$ are compatible if, for each color $c' \in \text{Col}$, we have $|f_1^{-1}(c')| = |f_2^{-1}(c')|$. Obviously, compatibility is an equivalence relation on token placements, and its equivalence classes are called compatibility classes for $G$ and $\text{Col}$. For a compatibility class $P$ and a cycle set $C$, we define the token-shifting graph of $P$ and $C$ as the undirected graph with vertex set $P$, where there is an edge between two token placements if and only if they are adjacent with respect to $C$. A walk in a token-shifting graph starting from $f$ and ending in $f'$ is called a shifting sequence between $f$ and $f'$, and the distance between $f$ and $f'$, i.e., the length of a shortest walk between them, is denoted as $\text{dist}(f, f')$ (if there is no walk between $f$ and $f'$, their distance is defined to be $\infty$). If $\text{dist}(f, f') < \infty$, we write $f \simeq f'$.

For a given number of colors $c$, we define the $c$-Colored Token Shift problem as follows. The input is a graph $G = (V, E)$, a cycle set $C$ for $G$, two compatible token placements $f_0$ and $f_1$ (with colors drawn from the set $\text{Col} = \{1, 2, \ldots, c\}$), and a non-negative integer $\ell$. The goal is to determine whether $\text{dist}(f_0, f_1) \leq \ell$ holds. In the case that $\ell$ is not given, we consider the $c$-Colored Token Shift problem as an optimization problem that aims at computing $\text{dist}(f_0, f_1)$.

3 Algebraic Analysis of the Puzzles

For the purpose of this section, the vertex set of the graph $G = (V, E)$ will be $V = \{1, 2, \ldots, n\}$, and the number of colors will be $c = n$, so that $\text{Col} = V$, and a token placement on $G$ can be interpreted as a permutation of $V$. To denote a permutation $\pi$ of $V$, we can either use the one-line notation $\pi = [\pi(1) \pi(2) \ldots \pi(n)]$, or we can write down its cycle decomposition: for instance, the permutation $[3 \ 6 \ 4 \ 1 \ 7 \ 2 \ 5]$ can be expressed as the product of disjoint cycles $(1 \ 3 \ 4)(2 \ 6)(5 \ 7)$.

Note that, given a cycle set $C$, shifting tokens along a cycle $(v_1, v_2, \ldots, v_j) \in C$ corresponds to applying the permutation $(v_1 \ v_2 \ldots \ v_j)$ or its inverse $(v_j \ v_{j-1} \ldots \ v_1)$
to $V$. The set of token placements generated by shifting sequences starting from the “identity token placement” $f_0 = [1 \ 2 \ \ldots \ n]$ is therefore a permutation group with the composition operator, which we denote by $H_C$, and we call it configuration group generated by $C$. Since we visualize permutations as functions mapping vertices of $G$ to colors (and not the other way around), it makes sense to compose chains of permutations from right to left, contrary to the common convention in the permutation group literature. So, for example, if we start from the identity token placement for $n = 5$ and we shift tokens along the cycles $(1 \ 2 \ 3)$ and $(3 \ 4 \ 5)$ in this order, we obtain the token placement 

$$(1 \ 2 \ 3)(3 \ 4 \ 5) = [2 \ 3 \ 1 \ 4 \ 5] \ [1 \ 2 \ 4 \ 5 \ 3] = [2 \ 3 \ 4 \ 5 \ 1] = (1 \ 2 \ 3 \ 4 \ 5).$$

(Had we composed permutations from left to right, we would have obtained the token placement $[2 \ 4 \ 1 \ 5 \ 3] = (1 \ 2 \ 4 \ 5 \ 3)$ as a result.)

One of our goals in this section is to determine the configuration groups $H_C$ generated by some classes of cycle sets $C$. Our choice of $C$ will be inspired by the puzzles in Fig. 1, and will consist of arrangements of cycles that share either one or two adjacent vertices. As we will see, except in one special case, the configuration groups that we obtain are either the symmetric group $S_n$ (i.e., the group of all permutations) or the alternating group $A_n$ (i.e., the group of all even permutations), depending on whether the cycle set $C$ contains at least one even-length cycle or not: indeed, observe that a cycle of length $j$ corresponds to an even permutation if and only if $j$ is odd.

Note that the set of permutations in the configuration group $H_C$ coincides with the connected component of the token-shifting graph (as defined in the previous section) that contains $f_0$. The other connected components are simply given by the cosets of $H_C$ in $S_n$ (thus, they all have the same size), while the number of connected components of the token-shifting graph is equal to the index of $H_C$ in $S_n$, i.e., $n!/|H_C|$.

The other goal of this section is to estimate the diameter of the token-shifting graph, i.e., the maximum distance between any two token placements $f_0$ and $f_t$ such that $f_0 \simeq f_t$. To this end, we state some basic preliminary facts, which are folklore, and can be proved by mimicking the “bubble sort” algorithm.

**Proposition 1.**

1. The $n$-cycle $(1 \ 2 \ \ldots \ n)$ and the transposition $(1 \ 2)$ can generate any permutation of $\{1, 2, \ldots, n\}$ in $O(n^2)$ shifts.
2. The $n$-cycle $(1 \ 2 \ \ldots \ n)$ and the 3-cycle $(1 \ 2 \ 3)$ can generate any even permutation of $\{1, 2, \ldots, n\}$ in $O(n^2)$ shifts.
3. The 3-cycles $(1 \ 2 \ 3)$, $(2 \ 3 \ 4)$, $\ldots$, $(n - 2 \ n - 1 \ n)$ can generate any even permutation of $\{1, 2, \ldots, n\}$ in $O(n^2)$ shifts.

All upper bounds given in Proposition 1 are worst-case asymptotically optimal (refer to \cite{6} for some proofs).

1 Of course, the two cycles generate strictly more than $A_n$ (hence $S_n$) if and only if $n$ is even; however, we will only apply Proposition 1.2 to generate even permutations.
3.1 Puzzles with two cycles

We first investigate the case where the cycle set $C$ contains exactly two cycles $\alpha$ and $\beta$, either of the form $\alpha = (1 \ 2 \ \ldots \ a)$ and $\beta = (a \ a+1 \ \ldots \ n)$ with $1 < a < n$, or of the form $\alpha = (1 \ 2 \ \ldots \ a)$ and $\beta = (a-1 \ a \ a+1 \ \ldots \ n)$, with $1 < a \leq n$.

The first puzzle is called $1$-connected $(a,b)$-puzzle, where $n = a + b - 1$, and the second one is called $2$-connected $(a,b)$-puzzle, where $n = a + b - 2$ (so, in both cases $a > 1$ and $b > 1$ are the lengths of the two cycles $\alpha$ and $\beta$, respectively). See Fig. 2 for some examples. Note that the Turnstile puzzle in Fig. 1 (left) can be regarded as a $2$-connected $(6,6)$-puzzle.

![Fig. 2. A 1-connected (5,7)-puzzle (left) and a 2-connected (6,9)-puzzle (right)](image)

**Theorem 1.** The configuration group of a $1$-connected $(a,b)$-puzzle is $A_n$ if both $a$ and $b$ are odd, and it is $S_n$ otherwise. Any permutation in the configuration group can be generated in $O(n^2)$ shifts.

**Proof.** Observe that the commutator of $\alpha$ and $\beta^{-1}$ is the $3$-cycle $\alpha^{-1}\beta\alpha\beta^{-1} = (a-1 \ a \ a+1)$. So, we can apply Proposition 1.2 to the $n$-cycle $\alpha\beta = (1 \ 2 \ \ldots \ n)$ and the $3$-cycle $(a-1 \ a \ a+1)$ to generate any even permutation in $O(n^2)$ shifts. If $a$ and $b$ are odd, then $\alpha$ and $\beta$ are even permutations, and therefore cannot generate any odd permutation.

On the other hand, if $a$ is even (the case where $b$ is even is symmetric), then the $a$-cycle $\alpha$ is an odd permutation. So, to generate any odd permutation $\pi \in S_n$, we first generate the even permutation $\pi\alpha$ in $O(n^2)$ shifts, and then we do one extra shift along the cycle $\alpha^{-1}$.

Our first observation about $2$-connected $(a, b)$-puzzles is that the composition of $\alpha^{-1}$ and $\beta$ is the $(n-1)$-cycle $\alpha^{-1}\beta = (a-2 \ a-3 \ \ldots \ 1 \ a \ a+1 \ \ldots \ n)$, which excludes only the element $a-1$. Similarly, $\alpha\beta^{-1} = (1 \ 2 \ \ldots \ a-1 \ n \ n-1 \ \ldots \ a+1)$, which excludes only the element $a$. We will write $\gamma_1$ and $\gamma_2$ as shorthand for $\alpha^{-1}\beta$ and $\alpha\beta^{-1}$, respectively.
and $\alpha\beta^{-1}$ respectively, and we will use the permutations $\gamma_1$ and $\gamma_2$ to conjugate $\alpha$ and $\beta$, thus obtaining different $a$-cycles and $b$-cycles.

**Lemma 1.** In a 2-connected $(3,b)$-puzzle, any even permutation can be generated in $O(n^2)$ shifts.

*Proof.* If we conjugate the 3-cycle $\alpha$ and $\gamma_1$ and $\alpha\beta$ and the conjugate of $\beta$ in Lemma 1. In a 2-connected

in a number of shifts independent of $\tau$, we can obtain the 3-cycle

by the inverse of $\gamma_1$, we obtain the 3-cycle $\gamma_1\alpha\alpha^{-1}\gamma_1 = (2\ 3\ 4)$. By applying Proposition 1.2 to the $(n-1)$-cycle $\beta$ and the 3-cycle $(2\ 3\ 4)$, we can generate any even permutation of $V\setminus\{1\}$ in $O(n^2)$ shifts.

Let $\pi \in A_n$ be an even permutation of $V$. In order to generate $\pi$, we first move the correct token $\pi(1)$ to position 1 in $O(n)$ shifts, possibly scrambling the rest of the tokens: let $\sigma$ be the resulting permutation. If $\sigma$ is even, then $\sigma^{-1}\pi$ is an even permutation of $V\setminus\{1\}$, and we can generate it in $O(n^2)$ shifts as shown before, obtaining $\pi$.

On the other hand, if $\sigma$ is odd, then one of the generators $\alpha$ and $\beta$ must be odd, too. Since $\alpha$ is a 3-cycle, it follows that $\beta$ is odd. In this case, after placing the correct token in position 1 via $\sigma$, we shift the rest of the tokens along $\beta$, and then we follow up with $\beta^{-1}\sigma^{-1}\pi$, which is an even permutation of $V\setminus\{1\}$, and can be generated in $O(n^2)$ shifts. Again, the result is $\sigma\beta\beta^{-1}\sigma^{-1}\pi = \pi$. □

**Lemma 2.** In a 2-connected $(a,b)$-puzzle with $a \geq 4$ and $b \geq 5$, any even permutation can be generated in $O(n^2)$ shifts.

*Proof.* As shown in Fig. 3, the conjugate of $\beta$ by $\gamma_1$ is the $b$-cycle

$$\delta_1 = \gamma_1^{-1}\beta\gamma_1 = (a\ a + 1 \ldots \ n - 1\ a - 1\ 1),$$

and the conjugate of $\beta^{-1}$ by $\gamma_2$ is the $b$-cycle

$$\delta_2 = \gamma_2^{-1}\beta^{-1}\gamma_2 = (n\ n - 1 \ldots \ a + 2\ a\ a - 2\ a - 1).$$

Their composition is $\delta_1\delta_2 = (1\ a\ a - 2)(a - 1\ n)(a + 1\ a + 2)$, and therefore $(\delta_1\delta_2)^2$ is the 3-cycle $(1\ a - 2\ a)$. Conjugating this 3-cycle by $\alpha^{-1}$, we finally obtain the 3-cycle $\tau = \alpha(\delta_1\delta_2)^2\alpha^{-1} = (1\ 2\ a - 1)$; note that $\tau$ has been generated in a number of shifts independent of $n$. Now, since the 3-cycle $\tau$ and the $(n-1)$-cycle $\gamma_2$ induce a 2-connected $(3, n-1)$-puzzle on $V$, we can apply Lemma 1 to generate any even permutation of $V$ in $O(n^2)$ shifts. □

**Theorem 2.** The configuration group of a 2-connected $(a,b)$-puzzle is:

1. Isomorphic to $S_{n-1} = S_5$ if $a = b = 4$.
2. $A_n$ if both $a$ and $b$ are odd.
3. $S_n$ otherwise.

Any permutation in the configuration group can be generated in $O(n^2)$ shifts.

---

2 If $g$ and $h$ are two elements of a group, the *conjugate* of $g$ by $h$ is defined as $h^{-1}gh$.

In the context of permutation groups, conjugation by any $h$ is an automorphism that preserves the cycle structure of permutations [9, Theorem 3.5].
Proof. By the symmetry of the puzzle, we may assume $a \leq b$. The case with $a = 2$ is equivalent to Proposition 1.1, so let $a \geq 3$. If $a \neq 4$ or $b \neq 4$, then Lemmas 1 and 2 apply, hence we can generate any even permutation in $O(n^2)$ shifts: the configuration group is therefore at least $A_n$. Now we reason as in Theorem 1 if $a$ and $b$ are odd, then $\alpha$ and $\beta$ are even permutations, and cannot generate any odd one. If $a$ is even (the case where $b$ is even is symmetric), then $\alpha$ is an odd permutation. In this case, to generate any odd permutation $\pi \in S_n$, we first generate the even permutation $\pi \alpha$ in $O(n^2)$ shifts, and then we do one more shift along the cycle $\alpha^{-1}$ to obtain $\pi$.

The only case left is $a = b = 4$. To analyze the 2-connected $(4,4)$-puzzle, consider the outer automorphism $\psi: S_6 \to S_6$ defined on a generating set of $S_6$ as follows (cf. [9, Corollary 7.13]):

$$
\psi((1 \ 2)) = (1 \ 5)(2 \ 3)(4 \ 6), \quad \psi((1 \ 3)) = (1 \ 4)(2 \ 6)(3 \ 5),
$$
$$
\psi((1 \ 4)) = (1 \ 3)(2 \ 4)(5 \ 6), \quad \psi((1 \ 5)) = (1 \ 2)(3 \ 6)(4 \ 5),
$$
$$
\psi((1 \ 6)) = (1 \ 6)(2 \ 5)(3 \ 4).
$$

Because $\psi$ is an automorphism, the subgroup of $S_6$ generated by $\alpha$ and $\beta$ is isomorphic to the subgroup generated by the permutations $\psi(\alpha)$ and $\psi(\beta)$. Since $\alpha = (1 \ 2 \ 3 \ 4) = (1 \ 2)(1 \ 3)(1 \ 4)$ and $\beta = (3 \ 4 \ 5 \ 6) = (1 \ 3)(1 \ 4)(1 \ 5)(1 \ 6)(1 \ 3)$, and recalling that $\psi(\pi_1 \pi_2) = \psi(\pi_1)\psi(\pi_2)$ for all $\pi_1, \pi_2 \in S_6$, we have:

$$
\psi(\alpha) = \psi((1 \ 2))\psi((1 \ 3))\psi((1 \ 4)) = [1 \ 5 \ 6 \ 4 \ 3 \ 2] = (2 \ 5 \ 3 \ 6) \text{ and}
$$
$$
\psi(\beta) = \psi((1 \ 3))\psi((1 \ 4))\psi((1 \ 5))\psi((1 \ 6))\psi((1 \ 3)) = [3 \ 1 \ 5 \ 4 \ 2 \ 6] = (1 \ 3 \ 5 \ 2).
$$

Note that the new generators $\psi(\alpha)$ and $\psi(\beta)$ both leave the token 4 in place, and so they cannot generate a subgroup larger than $S_5$ (up to isomorphism). On the other hand, we have $\psi(\alpha)\psi(\beta) = (1 \ 6 \ 2)$. This 3-cycle, together with the 4-cycle $\psi(\alpha)$, induces a 2-connected $(3,4)$-puzzle on $\{1, 2, 3, 5, 6\}$: as shown before, the configuration group of this puzzle is (isomorphic to) $S_5$. We conclude that the configuration group of the 2-connected $(4,4)$-puzzle is isomorphic to $S_5$, as well. A given permutation $\pi \in S_6$ is in the configuration group if and only if $\psi(\pi)$ leaves the token 4 in place. \qed
3.2 Puzzles with any number of cycles

Let us generalize the \((a,b)\)-puzzle to larger numbers of cycles. (As far as the authors know, there are commercial products that have 2, 3, 4, and 6 cycles.) We say that two cycles are \textit{properly interconnected} if they share exactly one vertex, or if they share exactly two vertices which are consecutive in both cycles. 

Note that all 1-connected and 2-connected \((a,b)\)-puzzles consist of two properly interconnected cycles. Given a set of cycles \(\mathcal{C}\) in a graph \(G = (V,E)\), let us define the \textit{interconnection graph} \(\hat{G} = (\mathcal{C}, \hat{E})\), where there is an (undirected) edge between two cycles of \(\mathcal{C}\) if and only if they are properly interconnected.

Let us assume \(|V| > 6\) (to avoid special configurations of small size, which can be analyzed by hand), and let \(\mathcal{C}\) consist of \(k\) cycles of lengths \(n_1, n_2, \ldots, n_k\), respectively. We say that \(\mathcal{C}\) induces a \textit{generalized} \((n_1,n_2,\ldots,n_k)\)-puzzle on \(V\) if there is a subset \(\mathcal{C}' \subseteq \mathcal{C}\) such that:

1. \(\mathcal{C}'\) contains at least two cycles;
2. the induced subgraph \(\hat{G}[\mathcal{C}']\) is connected;
3. each vertex of \(G\) is contained in at least one cycle in \(\mathcal{C}'\).

When we fix such a subset \(\mathcal{C}'\), the cycles in \(\mathcal{C}'\) are called \textit{relevant cycles}, and the vertices of \(G\) that are shared by two properly interconnected relevant cycles are called \textit{relevant vertices} for those cycles. See Fig. 4 for an example of a generalized puzzle.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{generalized_puzzle.png}
\caption{A generalized puzzle where any permutation can be generated in \(O(n^5)\) shifts, due to Theorem 3. Note that the blue cycle is the only cycle of even length, and is not properly interconnected with any other cycle. Also, the two red cycles and the two green cycles intersect each other but are not properly interconnected.}
\end{figure}

The next two lemmas are technical; their proof is found in the Appendix.

\textbf{Lemma 3.} In a generalized puzzle with three relevant cycles, \(\mathcal{C}' = \{C_1, C_2, C_3\}\), such that \(C_1\) and \(C_2\) induce a 2-connected \((4,4)\)-puzzle, any permutation involving only vertices in \(C_1\) and \(C_2\) can be generated in \(O(n^2)\) shifts. \(\square\)
Lemma 4. Let $V = \{1, \ldots, n\}$, and let $W = (w_1, \ldots, w_m) \in \mathbb{V}^m$ be a sequence such that each element of $V$ appears in $W$ at least once, and any three consecutive elements of $W$ are distinct. Then, the set of 3-cycles $C = \{(w_{i-1}, w_i, w_{i+1}) \mid 1 \leq i < m\}$ can generate any even permutation of $V$ in $O(n^3)$ shifts.

Theorem 3. The configuration group of a generalized $(n_1, n_2, \ldots, n_k)$-puzzle is $A_n$ if $n_1, n_2, \ldots, n_k$ are all odd, and it is $S_n$ otherwise. Any permutation in the configuration group can be generated in $O(n^3)$ shifts.

Proof. Observe that it suffices to prove that the given cycles can generate any even permutation in $O(n^3)$ shifts. Indeed, if all cycles have odd length, they cannot generate any odd permutation. On the other hand, if there is a cycle of even length and we want to generate an odd permutation $\pi$, we can shift tokens along that cycle, obtaining an odd permutation $\pi$, and then we can generate the even permutation $\pi^{-1} \pi$ in $O(n^3)$ shifts, obtaining $\pi$.

Let us fix a set of $k' \geq 3$ relevant cycles $C' \subseteq C$: we will show how to generate any even permutation by shifting tokens only along relevant cycles. By properties (2) and (3) of generalized puzzles, there exists a walk $W$ on $G$ that visits all vertices (possibly more than once), traverses only edges of relevant cycles, and transitions from one relevant cycle to another only if they are properly interconnected, and only through a relevant vertex shared by them. We will now slightly modify $W$ so that it satisfies the hypotheses of Lemma 4 as well as some other conditions. Namely, if $w_{i-1}, w_i, w_{i+1}$ are any three vertices that are consecutive in $W$, we would like the following conditions to hold:

1. $w_{i-1}, w_i, w_{i+1}$ are all distinct (this is the condition required by Lemma 4);
2. either $w_{i-1}$ and $w_i$ are in the same relevant cycle, or $w_i$ and $w_{i+1}$ are in the same relevant cycle;
3. $w_{i-1}$ and $w_{i+1}$ are either in the same relevant cycle, or in two properly interconnected relevant cycles.

To satisfy all conditions, it is sufficient to let $W$ do a whole loop around a relevant cycle before transitioning to the next (note that Lemma 4 applies regardless of the length of $W$). The only case where this is not possible is when $W$ has to go through a relevant 2-cycle $C = (u_1 u_2)$ that is a leaf in the induced subgraph $G[C']$, such that $C$ shares exactly one relevant vertex, say $u_1$, with another relevant cycle $C' = (v_0 u_1 v_1 v_2 \ldots)$. To let $W$ cover $C$ in a way that satisfies the above conditions, we set either $W = (\ldots, v_0, u_1, u_2, v_1, \ldots)$ or $W = (\ldots, v_1, u_1, v_2, v_0, \ldots)$; that is, we skip $u_1$ after visiting $u_2$. After this modification, $W$ is no longer a walk on $G$, but it satisfies the hypotheses of Lemma 4 as well as the three conditions above.

We will now show that the 3-cycle $(w_{i-1} w_i w_{i+1})$ can be generated in $O(n^2)$ shifts, for all $1 < i < |W|$. By Lemma 4 we will therefore conclude that any even permutation of $V$ can be generated in $O(n^2) \cdot O(n^3) = O(n^5)$ shifts. Due to conditions (2) and (3), we can assume without loss of generality that $w_{i-1}$ and $w_i$ are both in the same relevant cycle $C_1$, and that $w_{i+1}$ is either in $C_1$ or in a different relevant cycle $C_2$ which is properly interconnected with $C_1$. In the
first case, by property (1) of generalized puzzles, there exists another relevant cycle \( C_2 \) properly interconnected with \( C_1 \). So, in all cases, \( C_1 \) and \( C_2 \) induce a 1-connected or a 2-connected \(((C_1 \cup C_2)\)-puzzle.

That the 3-cycle \((w_{i-1}, w_i, w_{i+1})\) can be generated in \( O(n^2) \) shifts now follows directly from Theorems 1 and 2 except if \( |C_1| = |C_2| = 4 \) and \( C_1 \) and \( C_2 \) share exactly two vertices; indeed, the 2-connected \((4,4)\)-puzzle is the only case where we cannot generate any 3-cycle. However, since we are assuming that \( V > 6 \), there must be a third relevant cycle \( C_3 \), which is properly interconnected with \( C_1 \) or \( C_2 \). Our claim now follows from Lemma 3.

4 NP-Hardness for Puzzles with Two Colors

In this section, we show that the 2-Colored Token Shift problem is NP-hard. That is, for a graph \( G = (V,E) \), cycle set \( C \), two token placements \( f_0 \) and \( f_1 \) for \( G \), and a non-negative integer \( \ell \), it is NP-hard to determine if \( \text{dist}(f_0,f_1) \leq \ell \).

**Theorem 4.** The 2-Colored Token Shift problem is NP-hard.

**Proof.** We will give a polynomial-time reduction from the NP-complete problem 3-Dimensional Matching, or 3DM [3]: given three disjoint sets \( X, Y, Z \), each of size \( m \), and a set of triplets \( T \subseteq X \times Y \times Z \), does \( T \) contain a matching, i.e., a subset \( M \subseteq T \) of size exactly \( m \) such that all elements of \( X, Y, Z \) appear in \( M \).

Given an instance of 3DM \((X,Y,Z,T)\), with \( n = |T| \), we construct the instance of the 2-Colored Token Shift problem illustrated in Fig. 5.

The vertex set of \( G = (V,E) \) includes the sets \( X, Y, Z \) (shown with a green background in the figure: these will be called green vertices), as well as the vertex \( u \). Also, for each triplet \( t_i = (x_i,y_i,z_i) \in T \), with \( 1 \leq i \leq n \), the vertex set contains three vertices \( t_{i,1}, t_{i,2}, t_{i,3} \) (shown with a yellow background in the figure: these will be called yellow vertices), and the cycle set \( C \) has the three cycles \((u,t_{i,1},t_{i,2},t_{i,3},x_i)\), \((u,t_{i,1},t_{i,2},t_{i,3},y_i)\), and \((u,t_{i,1},t_{i,2},t_{i,3},z_i)\) (drawn in blue in the figure). Finally, we have the vertex \( w \), and the vertices \( v_1, v_2, \ldots, v_{3n-3m} \); for each \( i \in \{1,2,\ldots,n\} \), the cycle set \( C \) contains the cycle \((t_{i,3},t_{i,2},t_{i,1},v_1,v_2,\ldots,v_{3n-3m},w)\) (drawn in red in the figure). In the initial token placement \( f_0 \), there are black tokens on the \( 3n \) vertices of the form \( t_{i,j} \), and white tokens on all other vertices. In the final token placement \( f_1 \), there is a total of \( 3m \) black tokens on all the vertices in \( X, Y, Z \), plus \( 3n - 3m \) black tokens on \( v_1, v_2, \ldots, v_{3n-3m} \); all other vertices have white tokens. With this setup, we let \( \ell = 3n \).

It is easy to see that, if the 3DM instance has a matching \( M = \{t_{i_1}, t_{i_2}, \ldots, t_{i_m}\} \), then \( \text{dist}(f_0,f_1) \leq \ell \). Indeed, for each \( t_{i,j} = (x_j, y_j, z_j) \), with \( 1 \leq j \leq m \), we can shift tokens along the three blue cycles containing the yellow vertices \( t_{i,j,1}, t_{i,j,2}, t_{i,j,3} \), thus moving their three black tokens into the green vertices \( x_j, y_j, \) and \( z_j \). Since \( M \) is a matching, these \( 3m \) shifts eventually result in \( X, Y, \) and \( Z \) being covered by black tokens. Finally, we can shift the \( 3n - 3m \) black tokens corresponding to triplets in \( T \setminus M \) along red cycles, moving them into the vertices \( v_1, \ldots,
Fig. 5. The initial token placement $f_0$ (left) and the final token placement $f_t$ (right)

$v_2, \ldots, v_{3n-3m}$. Clearly, this is a shifting sequence of length $3n = \ell$ from $f_0$ to $f_t$.

We will now prove that, assuming that $\text{dist}(f_0, f_t) \leq \ell$, the 3DM instance has a matching. Note that each shift, no matter along which cycle, can move at most one black token from a yellow vertex to a non-yellow vertex. Since in $f_0$ there are $\ell = 3n$ black tokens on yellow vertices, and in $f_t$ no token is on a yellow vertex, it follows that each shift must cause exactly one black token to move from a yellow vertex to a non-yellow vertex, and no black token to move back into a yellow vertex.

This implies that no black token should ever reach vertex $u$: if it did, it would eventually have to be moved to some other location, because $u$ does not hold a black token in $f_t$. However, the black token in $u$ cannot be shifted back into a yellow vertex, and therefore it will be shifted into a green vertex along a blue cycle. Since every shift must cause a black token to leave the set of yellow vertices, such a token will move into $u$: we conclude that $u$ will always contain a black token, which is a contradiction. Similarly, we can argue that the vertex $w$ should never hold a black token.

Let us now focus on a single triplet of yellow vertices $t_{i,1}, t_{i,2}, t_{i,3}$. Exactly three shifts must involve these vertices, and they must result in the three black tokens leaving such vertices. Clearly, this is only possible if the three black tokens are shifted in the same direction. If they are shifted in the direction of $t_{i,3}$ (i.e., rightward in Fig. 5), they must move into green vertices (because they cannot
go into $u$); if they are shifted in the direction of $t_{i,1}$ (i.e., leftward in Fig. 5), they must move into $v_1$ (because they cannot go into $u$).

Note that, if a black token ever reaches a green vertex, it can no longer be moved; any shift involving such a token would move it back into a yellow vertex or into $u$. It follows that the only way of filling all the green vertices with black token is to select a subset of exactly $m$ triplets of yellow vertices and shift each of their black tokens into a different green vertex. These $m$ triplets of yellow vertices correspond to a matching for the 3DM instance.

In the above reduction, we can easily observe that the final token placement $f_t$ can always be reached from the initial token placement $f_0$ in a polynomial number of shifts. Therefore, for this particular set of instances, the 2-Colored Token Shift problem is in NP. The same is also true of the puzzles introduced in Section 3 due to the polynomial upper bound given by Theorem 3. However, we do not know whether this is true for the $c$-Colored Token Shift problem in general, even assuming $c = 2$. A theorem of Helfgott and Seress [5] implies that, if $f_0 \simeq f_t$, the distance between $f_0$ and $f_t$ has a quasi-polynomial upper bound; this, however, is insufficient to conclude that the problem is in NP. On the other hand, it is not difficult to see that the $c$-Colored Token Shift problem is in PSPACE; characterizing its computational complexity is left as an open problem. It would also be interesting to establish if the problem remains NP-hard when restricted to planar graphs or to graphs of constant maximum degree.

References

1. László Babai. The Probability of Generating the Symmetric Group. *Journal of Combinatorial Theory* (Series A), 52:148–153, 1989.
2. Merrick Furst, John Hopcroft, and Eugene Luks. Polynomial-Time Algorithms for Permutation Groups. In *Proceedings of the 21st Annual Symposium on Foundations of Computer Science*, 36–41, 1980.
3. Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.
4. Daniel Heath, I. M. Isaacs, John Kiltinen, and Jessica Sklar. Symmetric and Alternating Groups Generated by a Full Cycle and Another Element. *The American Mathematical Monthly*, 116(5):447–451, 2009.
5. Harald A. Helfgott and Ákos Seress. On the Diameter of Permutation Groups. *Annals of Mathematics*, 179(2):611–658, 2014.
6. Mark R. Jerrum. The Complexity of Finding Minimum-Length Generator Sequences. *Theoretical Computer Science*, 36:265–289, 1985.
7. Gareth A. Jones. Primitive Permutation Groups Containing a Cycle. *Bulletin of the Australian Mathematical Society*, 89(1):159–165, 2014.
8. Naomi Nishimura. Introduction to Reconfiguration. *Algorithms*, 11(4):1–25, 2018.
9. Joseph J. Rotman. *An Introduction to the Theory of Groups*. Springer-Verlag, 4th edition, 1995.
10. Richard M. Wilson. Graph Puzzles, Homotopy, and the Alternating Group. *Journal of Combinatorial Theory* (Series B), 16:86–96, 1974.
11. Katsuhisa Yamanaka, Takashi Horiyama, J. Mark Keil, David Kirkpatrick, Yota Otachi, Toshiki Saitoh, Ryuhei Uehara, and Yushi Uno. Swapping Colored Tokens on Graphs. *Theoretical Computer Science*, 729:1–10, 2018.
Appendix

Additional Figures

Fig. 6. Some cyclic shift puzzles with two (not properly interconnected) cycles

Fig. 7. More cyclic shift puzzles: Twiddler (left) and a puzzle found in the video game Haunted Manor 2 (right)

Missing Proofs

Lemma 3. In a generalized puzzle with three relevant cycles, $C' = \{C_1, C_2, C_3\}$, such that $C_1$ and $C_2$ induce a 2-connected $(4,4)$-puzzle, any permutation involving only vertices in $C_1$ and $C_2$ can be generated in $O(n^2)$ shifts.

Proof. Let $\alpha = (1\,2\,3\,4)$ and $\beta = (3\,4\,5\,6)$ be the permutations corresponding to shifting tokens along $C_1$ and $C_2$, respectively. As in Section 3.1 we set $\gamma_1 =$
\[ \alpha^{-1}\beta = (1\ 4\ 5\ 6\ 2) \text{ and } \gamma_2 = \alpha\beta^{-1} = (1\ 2\ 3\ 6\ 5). \] Since we are assuming that \(|V| > 6\), there must be a seventh vertex, and shifting along \(C_3\) corresponds to a permutation of the form \(\tau = (\ldots\ 7\ \ldots\).

We will prove that it is always possible to generate a transposition of the form (3 \(x\)), with \(x \in \{1, 2, 4, 5, 6\}\), in \(O(n^2)\) shifts. Indeed, such a transposition, together with the 5-cycle \(\gamma_1\), induces a 1-connected (2, 5)-puzzle on \(\{1, 2, 3, 4, 5, 6\}\).

Our lemma will thus follow from Theorem 1 and the fact that, in a 1-connected (2, 5)-puzzle, the distance between any two token placements is bounded by a constant.

If \(|C_3| \neq 4\), or if \(C_3\) is 1-connected with \(C_1\) or \(C_2\), then the transposition (3 \(x\)) can be generated in \(O(n^2)\) shifts, due to Theorems 1 and 2. So, we may assume that \(|C_3| = 4\), and \(C_3\) is properly interconnected with \(C_2\) and shares exactly two vertices with it. Perhaps, \(C_3\) shares at least two vertices with \(C_1\), as well. The only possible configurations, up to symmetry, are the following:

1. \(\tau = (3\ 4\ 7\ 8)\). Then, \(\tau\) and \(\gamma_1\) induce a 1-connected (4, 5)-puzzle on \(V\), and can generate the transposition (3 \(y\)) by Theorem 1.
2. \(\tau = (5\ 6\ 7\ 8)\). Then, \(\tau\) and \(\gamma_2\) induce a 2-connected (4, 5)-puzzle on \(V\ \backslash \{4\}\), and can generate the transposition (3 \(z\)) by Theorem 2.
3. \(\tau = (1\ 7\ 3\ 4)\). In this case, (3 \(2\)) \(= \tau^{-2}\alpha\alpha\tau\).
4. \(\tau = (1\ 3\ 4\ 7)\). In this case, (3 \(4\)) \(= \alpha\beta^{-1}\alpha^{-1}\tau\beta\tau^2\).
5. \(\tau = (1\ 3\ 6\ 7)\). In this case, (3 \(5\)) \(= \beta^{-1}\alpha\tau^{-1}\alpha\beta\tau^2\).
6. \(\tau = (1\ 6\ 3\ 7)\). In this case, (3 \(1\)) \(= \alpha\beta\alpha^{-1}\beta\tau^{-1}\beta\tau\).
7. \(\tau = (2\ 6\ 3\ 7)\). In this case, (3 \(4\)) \(= \omega^2\tau\tau^2\).
8. \(\tau = (2\ 3\ 6\ 7)\). In this case, (3 \(1\)) \(= \tau^{-1}\beta^{-1}\alpha\beta\alpha^{-1}\tau\alpha\).

\[ \square \]

**Lemma 4.** Let \(V = \{1, \ldots, n\}\), and let \(W = (w_1, \ldots, w_m) \in V^m\) be a sequence such that each element of \(V\) appears in \(W\) at least once, and any three consecutive elements of \(W\) are distinct. Then, the set of 3-cycles \(C = \{(w_{i-1}, w_i, w_{i+1}) \mid 1 < i < m\}\) can generate any even permutation of \(V\) in \(O(n^3)\) shifts.

**Proof.** Let \(\mu: V \to \{1, \ldots, m\}\) be the function mapping each \(v \in V\) to the minimum index \(\mu(v)\) such that \(w_{\mu(v)} = v\). Let \(\pi = [\pi_1 \ldots \pi_n]\) be the permutation of \(V\) such that the sequence \((\mu(\pi_1), \ldots, \mu(\pi_n))\) is monotonically increasing.

We will prove by induction on \(i\) that \(C\) can generate any 3-cycle on \(\{\pi_1, \ldots, \pi_i\}\) in at most 3\(i\) shifts. Assume this claim to be true up to a certain \(i < n\), and let us prove it for \(i + 1\). Let \(T = \{(\pi_j, \pi_{j'}, \pi_{i+1}) \mid 1 \leq j < j' \leq i\}\), and note that it suffices to prove that \(C\) generates all 3-cycles in \(T\), because the 3-cycles on \(\{\pi_{i+1}, \ldots, \pi_i\}\) are already accounted for by the inductive hypothesis.

So, fix one such 3-cycle \(\sigma_1 = (\pi_j, \pi_{j'}, \pi_{i+1}) \in T\), and observe that \(C\) already contains a 3-cycle in \(T\), namely \(\sigma_2 = (w_{\mu(\pi_{i+1})-2}, w_{\mu(\pi_{i+1})-1}, w_{\mu(\pi_{i+1})})\). Indeed, we have \(w_{\mu(\pi_{i+1})} = \pi_{i+1}\), and, by the minimality of \(\mu\), there exist two distinct indices \(k, k' \in \{1, \ldots, i\}\) such that \(w_{\mu(\pi_{i+1})-2} = \pi_k\) and \(w_{\mu(\pi_{i+1})-1} = \pi_{k'}\).

If \(\{j, j'\} = \{k, k'\}\), then \(\sigma_1 = \sigma_2\), and we are done. If \(\{j, j'\}\) and \(\{k, k'\}\) are disjoint, consider the 3-cycle \(\sigma_3 = (\pi_j, \pi_{j'}, \pi_k)\), which, by the inductive hypothesis, can be generated by \(C\) in at most 3\(i\) shifts. We have \(\sigma_1 = \sigma_2\sigma_3\sigma_2\sigma_2\), and so \(C\) can generate \(\sigma_1\) in at most 3\(i + 3 = 3(i + 1)\) shifts.
Otherwise, \( \{j, j'\} \) and \( \{k, k'\} \) intersect in exactly one element, which we may assume to be \( j' = k' \), without loss of generality. In this case, \( \sigma_1 = \sigma_2 \sigma_3 \), where \( \sigma_3 \) is defined as above. So, \( \mathcal{C} \) can generate \( \sigma_1 \) in at most \( 3i + 1 < 3(i + 1) \) shifts.

By taking \( i = n \), we conclude that \( \mathcal{C} \) can generate any 3-cycle on \( V \) in at most \( 3n = O(n) \) shifts, implying that it can generate any even permutation of \( V \) in \( O(n^3) \) shifts, due to Proposition 1.3. \( \square \)