PAPER *Special Section on Foundations of Computer Science - New Trends of Theory of Computation and Algorithm -*

Cyclic Shift Problems on Graphs*

Kwon Kham SAI[†a)], ***Nonmember***, **Giovanni VIGLIETTA**[†b)], ***and*** **Ryuhei UEHARA**[†c)], ***Members***

SUMMARY 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 call this a *cyclic shift puzzle*. We first investigate a large class of graphs, which generalizes several classic cyclic shift 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.

key words: *reconfiguration problem, cyclic shift puzzle, permutation group, generator, 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 $\textsc{Col} = \{1, 2, \dots, c\}$, and one colored token on each vertex in V. We are also given a set $\mathcal{C}$ of cycles of G. The basic operation on G is called "shift" along a cycle C in $\mathcal{C}$, 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. [12], and has been well investigated recently. Indeed, when we restrict each cycle in $\mathcal{C}$ to have length two (each cycle would correspond to an edge in E), the cyclic shift problem is equivalent to the token swapping problem.

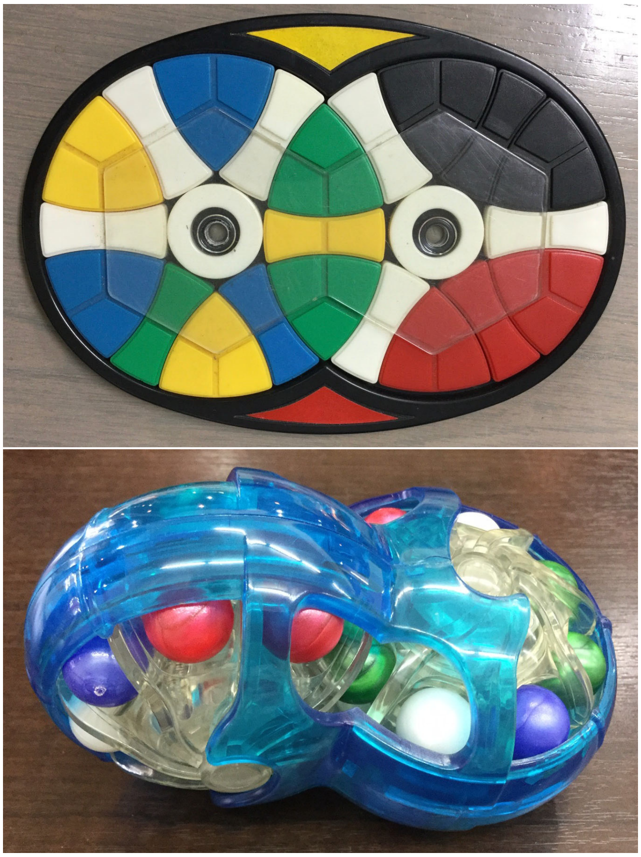


Fig. 1 Commercial cyclic shift puzzles: Turnstile (top) and Rubik's Shells (bottom)

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 $1 - 1/n + O(n^2)$. 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-

Manuscript received March 26, 2021.
Manuscript publicized October 8, 2021.

[†] The authors are with School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Nomi-shi, 923–1292 Japan.

[*] A preliminary version of this paper appeared in WALCOM 2021.

a) E-mail: saikwonkham@jaist.ac.jp
b) E-mail: johnny@jaist.ac.jp (Corresponding author)
c) E-mail: uehara@jaist.ac.jp

DOI: 10.1587/transinf.2021FCP0010

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 π in the group is decidable in polynomial time, and an expression for π in terms of the generators is also computable in polynomial time [2].

By 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 (Sect. 2), we study the configuration space of a large class of cyclic shift problems which generalize the puzzles in Fig. 1 (Sect. 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 [11]. Moreover, our proofs in Sect. 3 are constructive, and yield polynomial upper bounds on the number of shift operations required to reach a given configuration. This is contrasted with Sect. 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. We conclude the paper with some open problems (Sect. 5).

A preliminary version of this paper appeared in WALCOM 2021 [10]. This extended version contains missing proofs and additional figures.

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 $\textsc{Col} = \{1, 2, \dots, c\}$ be a set of colors, where c is a constant. A *token placement* for G is a function $f\colon V \to \textsc{Col}$: 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 $\mathcal{C}$ of cycles in G (note that $\mathcal{C}$ does not necessarily contain all cycles of G)[†]. Two distinct token placements f and f' of G are *adjacent* with respect to $\mathcal{C}$ if the following two conditions hold: (1) there exists a cycle $C = (v_1, v_2, \dots, v_j)$ in $\mathcal{C}$ 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 $w \in V \setminus \{v_1, \dots, v_i\}$. 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 $\mathcal{C}$, 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 $\mathcal{C}$.

We say that two token placements f_1 and f_2 are *compatible* if, for each color $c' \in \textsc{Col}$, we have $\left|f_1^{-1}(c')\right| = \left|f_2^{-1}(c')\right|$. Obviously, compatibility is an equivalence relation on token placements, and its equivalence classes are called *compatibility classes* for G and $\textsc{Col}$. For a compatibility class P and a cycle set $\mathcal{C}$, we define the *token-shifting graph of P and $\mathcal{C}$* 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 $\mathcal{C}$. 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 $\mathrm{dist}(f, f')$ (if there is no walk between f and f', their distance is defined to be ∞). If $\mathrm{dist}(f, f') < \infty$, 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 $\mathcal{C}$ for G, two compatible token placements f_0 and f_t (with colors drawn from the set $\textsc{Col} = \{1, 2, \dots, c\}$), and a non-negative integer ℓ. The goal is to determine whether $\mathrm{dist}(f_0, f_t) \leq \ell$ holds. In the case that ℓ is not given, we consider the c-Colored Token Shift problem as an optimization problem that aims at computing $\mathrm{dist}(f_0, f_t)$.

3. Algebraic Analysis of the Puzzles

For the purpose of this section, the vertex set of the graph $G = (V, E)$ will be $V = \{1, 2, \dots, n\}$, and the number of colors will be $c = n$, so that $\textsc{Col} = V$, and a token placement on G can be interpreted as a permutation of V. To denote a permutation π of V, we can either use the one-line notation $\pi = [\pi(1)\ \pi(2)\ \dots\ \pi(n)]$, 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 $\mathcal{C}$, shifting tokens along a cycle $(v_1, v_2, \dots, v_j) \in \mathcal{C}$ corresponds to applying the permutation $(v_1\ v_2\ \dots\ v_j)$ or its inverse $(v_j\ v_{j-1}\ \dots\ v_1)$ to V. The set of token placements generated by shifting sequences starting from the "identity token placement" $f_0 = [1\ 2\ \dots\ n]$ is therefore a permutation group with the composition operator, which we denote by $H_{\mathcal{C}}$, and we call it *configuration group generated by $\mathcal{C}$*. 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

$$\begin{aligned}(1\ 2\ 3)(3\ 4\ 5) &= [2\ 3\ 1\ 4\ 5]\,[1\ 2\ 4\ 5\ 3]\\ &= [2\ 3\ 4\ 5\ 1] = (1\ 2\ 3\ 4\ 5).\end{aligned}$$

[†] In this paper, we regard a pair of adjacent vertices as a cycle of length 2.

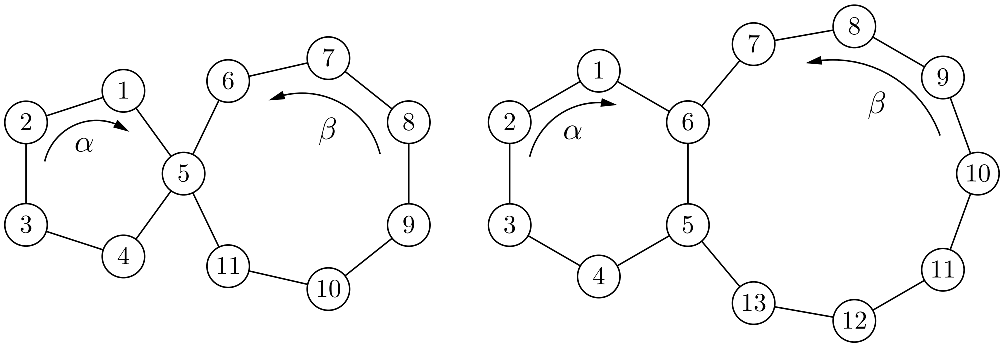


Fig. 2 A 1-connected $(5, 7)$-puzzle (left) and a 2-connected $(6, 9)$-puzzle (right)

(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 [6] for some proofs).

3.1 Puzzles with Two Cycles

We first investigate the case where the cycle set C contains exactly two cycles α and β, 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 α and β, respectively). See Fig. 2 for some examples. Note that the Turnstile puzzle in Fig. 1 (top) can be regarded as a 2-connected $(6, 6)$-puzzle.

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 α and β^{-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 α and β 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 α 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 α^{-1}. □

Our first observation about 2-connected (a, b)-puzzles is that the composition of α^{-1} and β 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 γ_1 and γ_2 as shorthand for $\alpha^{-1}\beta$ and $\alpha\beta^{-1}$ respectively, and we will use the permutations γ_1 and γ_2 to conjugate α and β, thus obtaining different a-cycles and b-cycles.[††]

[†] 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.

[††] 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].

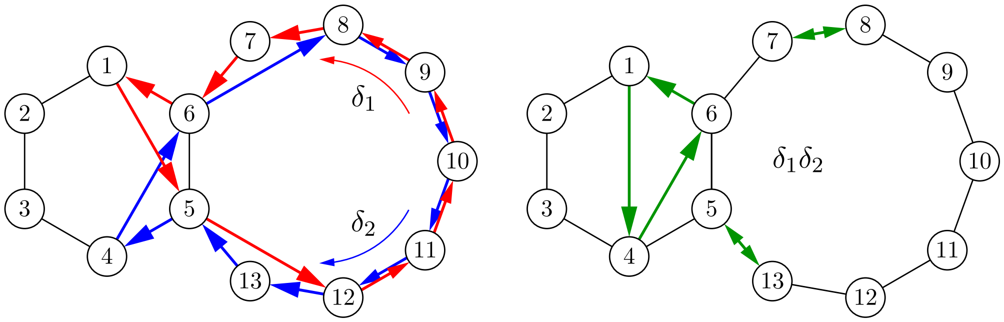


Fig. 3 Some permutations constructed in the proof of Lemma 2

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 α^{-1} by the inverse of γ_1, we obtain the 3-cycle $\gamma_1\alpha^{-1}\gamma_1^{-1} = (2\ 3\ 4)$. By applying Proposition 1.2 to the $(n-1)$-cycle β 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 π, we first move the correct token $\pi(1)$ to position 1 in $O(n)$ shifts, possibly scrambling the rest of the tokens: let σ be the resulting permutation. If σ 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 π.

On the other hand, if σ is odd, then one of the generators α and β must be odd, too. Since α is a 3-cycle, it follows that β is odd. In this case, after placing the correct token in position 1 via σ, we shift the rest of the tokens along β, and then we follow up with $\beta^{-1}\sigma^{-1}\pi$, which is an even permutation of $V \setminus \{1\}$, and can be generated it 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 β by γ_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 β^{-1} by γ_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 α^{-1}, we finally obtain the 3-cycle $\tau = \alpha(\delta_1\delta_2)^2\alpha^{-1} = (1\ 2\ a-1)$; note that τ has been generated in a number of shifts independent of n. Now, since the 3-cycle τ and the $(n-1)$-cycle γ_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.

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 α and β are even permutations, and cannot generate any odd one. If a is even (the case where b is even is symmetric), then α 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 α^{-1} to obtain π.

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

$$\begin{aligned}
\psi((1\ 2)) &= (1\ 5)(2\ 3)(4\ 6),\\
\psi((1\ 3)) &= (1\ 4)(2\ 6)(3\ 5),\\
\psi((1\ 4)) &= (1\ 3)(2\ 4)(5\ 6),\\
\psi((1\ 5)) &= (1\ 2)(3\ 6)(4\ 5),\\
\psi((1\ 6)) &= (1\ 6)(2\ 5)(3\ 4).
\end{aligned}$$

Because ψ is an automorphism, the subgroup of S_6 generated by α and β 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:

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

Note that the new generators $\psi(\alpha)$ and $\psi(\beta)$ both leave the token 4 in place, and so they cannot generate a subgroup

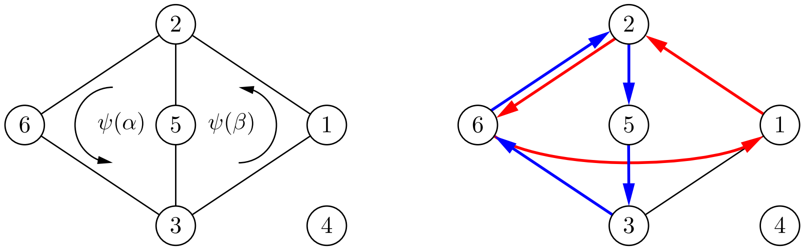


Fig. 4 Some permutations constructed in the proof of Theorem 2

larger than S_5 (up to isomorphism): Fig. 4 (left). On the other hand, we have $\psi(\alpha)\psi(\beta) = (1\ 6\ 2)$. This 3-cycle, together with the 4-cycle $\psi(\alpha)^{-1}$, induces a 2-connected $(3,4)$-puzzle on $\{1,2,3,5,6\}$: Fig. 4 (right). 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. □

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 *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 C in a graph $G = (V,E)$, let us define the *interconnection graph* $\hat{G} = (C,\hat{E})$, where there is an (undirected) edge between two cycles of 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 C consist of k cycles of lengths $n_1, n_2, \ldots, n_k$, respectively. We say that C induces a *generalized* $(n_1,n_2,\ldots,n_k)$*-puzzle* on V if there is a subset $C' \subseteq C$ such that:

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

When we fix such a subset C', the cycles in C' are called *relevant cycles*, and the vertices of G that are shared by two properly interconnected relevant cycles are called *relevant vertices* for those cycles. See Fig. 5 for an example of a generalized puzzle.

We first give two technical lemmas.

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 Sect. 3.1, we set $\gamma_1 = \alpha^{-1}\beta = (1\ 4\ 5\ 6\ 2)$ 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 $\tau = (\ldots\ 7\ \ldots)$.

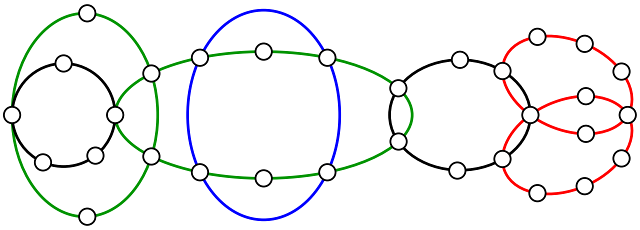


Fig. 5 A generalized puzzle where any permutation can be generated in $O(n^5)$ shifts, due to Theorem 3. Note that the blue cycle is the only cycle of even length, and is not properly interconnected with any other cycle. Also, the two red cycles and the two green cycles intersect each other but are not properly interconnected.

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 γ_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 $(3\ 4)$ 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, τ and γ_1 induce a 1-connected $(4,5)$-puzzle on V, and can generate the transposition $(3\ 4)$ by Theorem 1.
(2) $\tau = (5\ 6\ 7\ 8)$. Then, τ and γ_2 induce a 2-connected $(4,5)$-puzzle on $V \setminus \{4\}$, and can generate the transposition $(3\ 5)$ by Theorem 2.
(3) $\tau = (1\ 7\ 3\ 4)$. In this case, $(3\ 2) = \tau^{-2}\alpha\tau\alpha$.
(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, $(3\ 4) = \alpha^2\tau^2\alpha\tau^2$.
(8) $\tau = (2\ 3\ 6\ 7)$. In this case, $(3\ 1) = \tau^{-1}\beta^{-1}\alpha\beta\alpha^{-1}\tau\alpha$.

□

Lemma 4. *Let $V = \{1, \dots, n\}$, and let $W = (w_1, \dots, w_m) \in V^m$ 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 $\mu\colon V \to \{1, \dots, m\}$ be the function mapping each $v \in V$ to the minimum index $\mu(v)$ such that $w_{\mu(v)} = v$. Let $\pi = [\pi_1\ \dots\ \pi_n]$ be the permutation of V such that the sequence $(\mu(\pi_1), \dots, \mu(\pi_n))$ is monotonically increasing.

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

So, fix one such 3-cycle $\sigma_1 = (\pi_j\ \pi_{j'}\ \pi_{i+1}) \in \mathcal{T}$, and observe that C already contains a 3-cycle in $\mathcal{T}$, 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 μ, there exist two distinct indices $k, k' \in \{1, \dots, i\}$ 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 $3i$ shifts. We have $\sigma_1 = \sigma_2\sigma_3\sigma_2\sigma_2$, and so C can generate σ_1 in at most $3i + 3 = 3(i+1)$ 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 σ_3 is defined as above. So, C can generate σ_1 in at most $3i + 1 < 3(i+1)$ shifts.

By taking $i = n$, we conclude that 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. □

We are now in the position to prove the main theorem of this section.

Theorem 3. *The configuration group of a generalized $(n_1, n_2, \dots, n_k)$-puzzle is A_n if $n_1, n_2, \dots, n_k$ are all odd, and it is S_n otherwise. Any permutation in the configuration group can be generated in $O(n^5)$ shifts.*

Proof. Observe that it suffices to prove that the given cycles can generate any even permutation in $O(n^5)$ 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 π, we can shift tokens along that cycle, obtaining an odd permutation σ, and then we can generate the even permutation $\sigma^{-1}\pi$ in $O(n^5)$ shifts, obtaining π.

Let us fix a set of $k' \ge 3$ 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 $\hat{G}[C']$, such that C shares exactly one relevant vertex, say u_1, with another relevant cycle $C' = (v_0\ u_1\ v_1\ v_2\ \dots)$. To let W cover C in a way that satisfies the above conditions, we set either $W = (\dots, v_0, u_1, u_2, v_1, \dots)$ or $W = (\dots, v_1, u_1, u_2, v_0, \dots)$: 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 $(|C_1|, |C_2|)$-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_t for G, and a non-negative integer ℓ, it is NP-hard to determine if $\mathrm{dist}(f_0, f_t) \le \ell$.

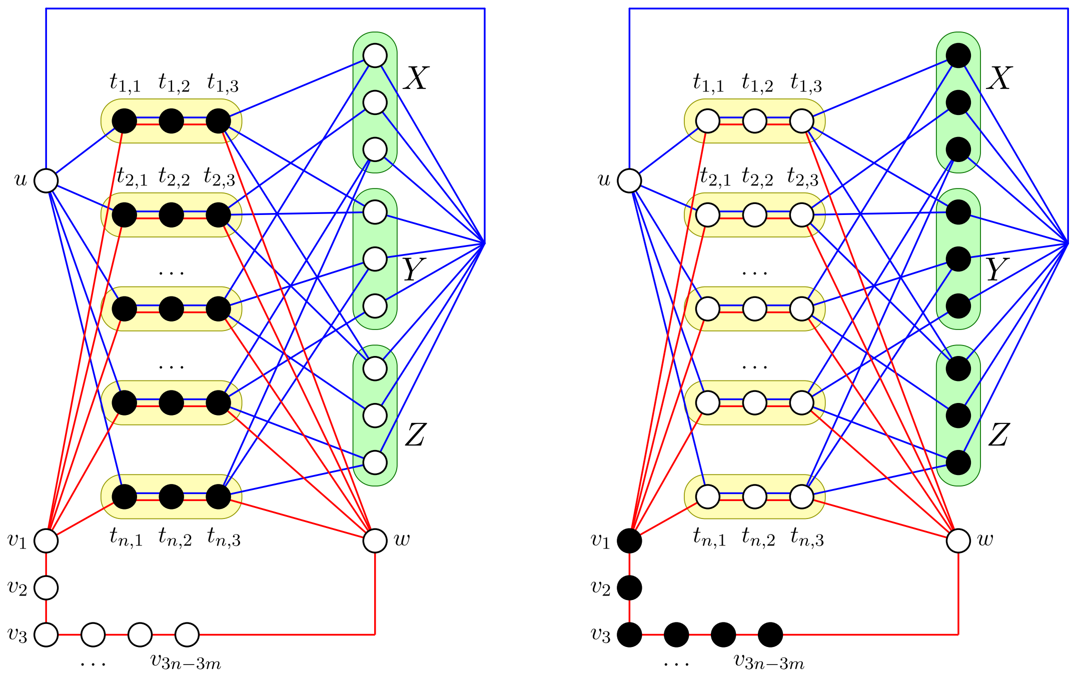

$t_{1,1}$ $t_{1,2}$ $t_{1,3}$
$t_{2,1}$ $t_{2,2}$ $t_{2,3}$
$t_{n,1}$ $t_{n,2}$ $t_{n,3}$
u
X
Y
Z
w
v_1
v_2
v_3
v_{3n-3m}
$t_{1,1}$ $t_{1,2}$ $t_{1,3}$
$t_{2,1}$ $t_{2,2}$ $t_{2,3}$
$t_{n,1}$ $t_{n,2}$ $t_{n,3}$
u
X
Y
Z
w
v_1
v_2
v_3
v_{3n-3m}

Fig. 6 The initial token placement f_0 (left) and the final token placement f_t (right)

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. 6.

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 $\hat{t}_i = (x, y, z) \in T$, with $1 \le i \le n$, 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 $(u, t_{i,1}, t_{i,2}, t_{i,3}, x)$, $(u, t_{i,1}, t_{i,2}, t_{i,3}, y)$, and $(u, t_{i,1}, t_{i,2}, t_{i,3}, z)$ (drawn in blue in the figure). Finally, we have the vertex w, and the vertices v_1, v_2, $\dots$, v_{3n-3m}; for each $i \in \{1, 2, \dots, n\}$, the cycle set C contains the cycle $(t_{i,3}, t_{i,2}, t_{i,1}, v_1, v_2, \dots, v_{3n-3m}, w)$ (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_t, 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, $\dots$, 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 $M = \{\hat{t}_{i_1}, \hat{t}_{i_2}, \dots, \hat{t}_{i_m}\}$, then $\operatorname{dist}(f_0, f_t) \le \ell$. Indeed, for each $\hat{t}_{i_j} = (x_j, y_j, z_j)$, with $1 \le j \le m$, 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, v_2, $\dots$, 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 $\operatorname{dist}(f_0, f_t) \le \ell$, 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. 6), they must move into green vertices (because they cannot go into w); if they are shifted in the direction of $t_{i,1}$ (i.e., leftward in Fig. 6), 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 tokens 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. □

5. Conclusion

We have introduced *cyclic shift puzzles* as a natural reconfiguration problem generalizing several classic mechanical puzzles. We have identified a large class of cyclic shift puzzles (based on the notion of *properly interconnected* cycles), and we have characterized their configuration spaces. Furthermore, we have given polynomial upper bounds on the number of shifts required to solve these puzzles. Finally, we have shown that solving a cyclic shift puzzle in the minimum number of shifts is NP-hard.

It would be interesting to extend our analysis of configuration spaces to cyclic shift puzzles where no cycles are properly interconnected, such as the ones in Fig. 7.

Also, in our NP-hardness 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 Sect. 3, due to the polynomal 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.

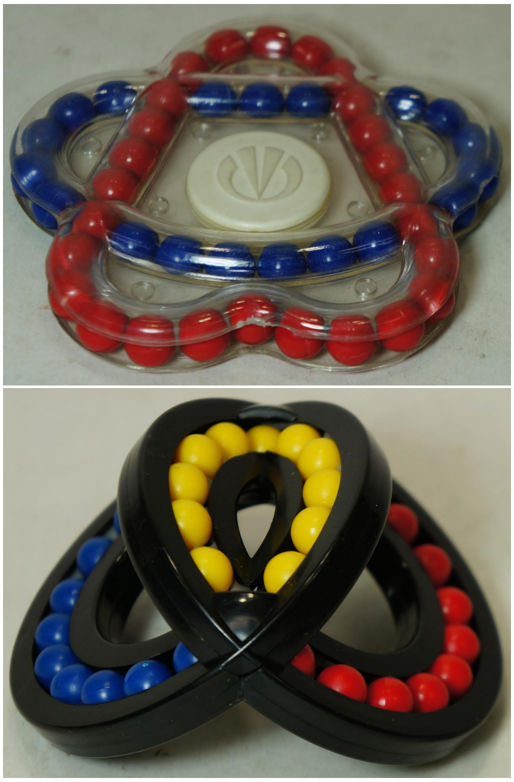


Fig. 7 Cyclic shift puzzles consisting of two cycles that are not properly interconnected

Acknowledgments

This work is partially supported by JSPS KAKENHI Grant Numbers 17H06287 and 18H04091.

References

[1] L. Babai, "The probability of generating the symmetric group," J. Combinatorial Theory (Series A), vol.52, pp.148–153, Sept. 1989.

[2] M. Furst, J. Hopcroft, and E. Luks, "Polynomial-time algorithms for permutation groups," Proc. 21st Annual Symposium on Foundations of Computer Science, pp.36–41, 1980.

[3] M.R. Garey and D.S. Johnson, " Computers and intractability: A guide to the theory of NP-completeness," W. H. Freeman and Company, 1979.

[4] D. Heath, I. M. Isaacs, J. Kiltinen, and J. Sklar, "Symmetric and alternating groups generated by a full cycle and another element," The American Mathematical Monthly, vol.116, no.5, pp.447–451, May 2009.

[5] H.A. Helfgott and Á. Seress, "On the diameter of permutation groups," Annals of Mathematics, vol.179, no.2, pp.611–658, 2014.

[6] M.R. Jerrum, The complexity of finding minimum-length generator sequences, Theoretical Computer Science, vol.36, pp.265–289, 1985.

[7] G.A. Jones, "Primitive permutation groups containing a cycle," Bulletin of the Australian Mathematical Society, vol.89, no.1, pp.159–165, Feb. 2014.

[8] N. Nishimura, "Introduction to reconfiguration," Algorithms, vol.11, no.4, pp.1–25, 2018.

[9] J.J. Rotman, An Introduction to the Theory of Groups, Springer-Verlag, 4th edition, 1995.

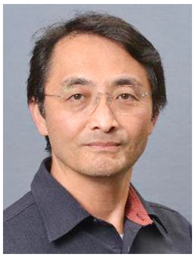
[10] K.K. Sai, R. Uehara, and G. Viglietta, "Cyclic shift problems on graphs," Proc. 15th International Conference and Workshop on Algorithms and Computation (WALCOM 2021), pp.308–320, 2021.

[11] R.M. Wilson, "Graph Puzzles, homotopy, and the alternating group," J. Combinatorial Theory (Series B), vol.16, no.1, pp.86–96, Feb. 1974.

[12] K. Yamanaka, T. Horiyama, J.M. Keil, D. Kirkpatrick, Y. Otachi, T. Saitoh, R. Uehara, and Y. Uno, "Swapping colored tokens on graphs," Theoretical Computer Science, vol.729, pp.1–10, June 2018.

Kwon Kham Sai received his Master's degree in Computer Science from the Japan Advanced Institute of Science and Technology (JAIST) in 2020 under the supervision of Prof. Ryuhei Uehara.

Giovanni Viglietta received his Ph.D. in Computer Science from the University of Pisa in 2013. He is now Assistant Professor at the Japan Advanced Institute of Science and Technology (JAIST).



Ryuhei Uehara is a professor in School of Information Science, JAIST. He received B.E., M.E., and Ph.D. degrees from the University of Electro-Communications, Japan, in 1989, 1991, and 1998, respectively. He was a researcher in CANON Inc. during 1991–1993. In 1993, he joined Tokyo Woman's Christian University as an assistant professor. He was a lecturer during 1998–2001, and an associate professor during 2001–2004 at Komazawa University. He moved to JAIST in 2004. His research interests include computational complexity, algorithms and data structures, and graph algorithms. Especially, he is engrossed in computational origami, games and puzzles from the viewpoints of theoretical computer science. He is a member of IPSJ and IEICE. He is the chair of Japan Chapter of EATCS.