


# Computational Complexity of Swish Is Solved

**Takashi Horiyama** ✉ 

Faculty of Information Science and Technology, Hokkaido University, Sapporo, Japan

**Takehiro Ito** ✉ 


Graduate School of Information Sciences, Tohoku University, Sendai, Japan

**Jun Kawahara** ✉ 

Graduate School of Informatics, Kyoto University, Kyoto, Japan

**Shin-ichi Minato** ✉ 

Graduate School of Informatics, Kyoto University, Kyoto, Japan

**Akira Suzuki** ✉ 

Center for Data-driven Science and Artificial Intelligence, Tohoku University, Sendai, Japan

**Ryuhei Uehara** ✉ 

School of Information Science, Japan Advanced Institute of Science and Technology, Nomi, Japan

**Yutaro Yamaguchi** ✉ 

Graduate School of Information Science and Technology, Osaka University, Suita, Japan

---

## Abstract

Swish is a card game in which players are given cards having symbols (hoops and balls), and find a valid superposition of cards, called a “swish.” Dailly, Lafourcade, and Marcadet (FUN 2024) studied a generalized version of Swish and showed that the problem is solvable in polynomial time with one symbol per card, while it is NP-complete with three or more symbols per card. In this paper, we resolve the previously open case of two symbols per card, which corresponds to the original game. We show that Swish is NP-complete for this case. Specifically, we prove the NP-hardness when the allowed transformations of cards are restricted to a single (horizontal or vertical) flip or 180-degree rotation, and extend the results to the original setting allowing all three transformations. In contrast, when neither transformation is allowed, we present a polynomial-time algorithm. Combining known and our results, we establish a complete characterization of the computational complexity of Swish with respect to both the number of symbols per card and the allowed transformations.

**2012 ACM Subject Classification** Theory of computation → Problems, reductions and completeness; Theory of computation → Design and analysis of algorithms; Theory of computation → Graph algorithms analysis

**Keywords and phrases** Swish, Computational complexity, Matching, Parity-constrained cycles

**Digital Object Identifier** 10.4230/LIPIcs.FUN.2026.25

**Funding** *General:* JSPS/MEXT KAKENHI Grant Numbers JP20H00605, JP20H05964, JP24H00690, JP25H01114

*Takashi Horiyama:* JSPS KAKENHI Grant Number JP23K24806.

*Takehiro Ito:* JSPS KAKENHI Grant Number JP24H00686.

*Akira Suzuki:* JSPS KAKENHI Grant Number JP25K14980.

*Yutaro Yamaguchi:* JST CRONOS Japan Grant Number JPMJCS24K2.

**Acknowledgements** We would like to thank the reviewers for enjoying the paper and giving helpful comments.

## 1 Introduction

SWISH [15] is a card game in which players try to find a valid superposition of cards. In the game, cards are overlaid so that every hoop (circle)  $\bigcirc$  meets a ball (point)  $\bullet$ , every ball meets a hoop, and neither two hoops nor two balls overlap; such a configuration is called a *swish*.



© Takashi Horiyama, Takehiro Ito, Jun Kawahara, Shin-ichi Minato, Akira Suzuki, Ryuhei Uehara, and Yutaro Yamaguchi;

licensed under Creative Commons License CC-BY 4.0

13th International Conference on Fun with Algorithms (FUN 2026).

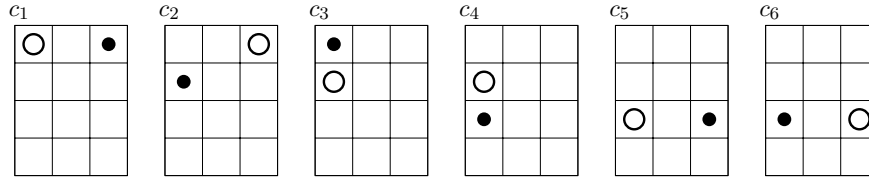
Editor: John Iacono; Article No. 25; pp. 25:1–25:12



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

## 25:2 Computational Complexity of Swish Is Solved



■ **Figure 1** Example of cards in SWISH. Following the notation introduced in Section 2.1,  $c_1 = ((1, 1), (1, 3))$ ,  $c_2 = ((1, 3), (2, 1))$ , etc. The swishes of this instance without rotation or flip are  $\{c_1, c_2, c_3\}$ ,  $\{c_5, c_6\}$ , and  $\{c_1, c_2, c_3, c_5, c_6\}$ . If we are allowed both flip and rotation,  $\{c_2, c_3, c_6\}$  is also a swish, where we use the cards as  $c_2 = ((1, 3), (2, 1))$ ,  $c_3^F = ((2, 3), (1, 3))$ , and  $c_6^R = ((2, 1), (2, 3))$ , where the superscripts F and R mean horizontal flip and 180-degree rotation, respectively.

The commercial version of this game consists of 60 transparent cards, each with exactly one hoop and exactly one ball arranged on a grid of height 4 and width 3 (see Figure 1 for examples). Cards may be flipped horizontally or vertically and rotated by 180 degrees, which yields 36 distinct cards up to symmetry; some (equivalent) cards therefore appear twice in the deck. In the commercial game, hoops and balls are colored in blue, orange, purple, and green, but these colors serve only to visually distinguish positions. From a combinatorial point of view, the game can be regarded as *monochromatic*, that is, symbols themselves carry no colors. In each round of the game, given 16 cards in common drawn from the deck, the players compete to find a swish faster than their opponents and call out “Swish!” Despite its simple rules, SWISH exhibits a rich combinatorial structure that makes it an appealing subject for mathematical and algorithmic analysis.

Rowland [12] initiated a mathematical study of SWISH, focusing on structural and probabilistic aspects of the commercial deck. Using Burnside’s theorem, she showed that the 60-card deck contains 36 distinct cards up to symmetry. She further analyzed the distribution of symbol positions, showing that they do not appear uniformly, and studied the probability of forming a swish with two cards. SWISH was also discussed from a linear-algebraic perspective via matrix representations.

From a computational complexity viewpoint, SWISH was first studied systematically by Dailly, Lafourcade, and Marcadet [3] in FUN 2024. They introduced a generalized version of SWISH and analyzed its complexity with respect to the number of symbols per card. They showed that SWISH can be solved in polynomial time when each card contains a single symbol, whereas the problem becomes NP-complete when cards may contain three or more symbols. To establish NP-completeness, they also considered a restricted variant in which cards cannot be flipped or rotated, often referred to as SIMPLE-SWISH.

In this paper, we complete the complexity classification of SWISH by addressing two aspects that remain open. The first concern is the number of symbols per card: what is the complexity of SWISH when each card contains exactly two symbols, as in the original game? The second concern is the role of geometric transformations: how does the complexity change when flip and rotation are selectively allowed or forbidden?

We show a sharp boundary. When neither flip nor rotation is allowed, SWISH with two symbols per card admits a polynomial-time algorithm (Theorem 1). In contrast, if even a single type of transformation (i.e., one of the horizontal flip, vertical flip, and 180-degree rotation) is allowed, the problem becomes NP-complete, already with one hoop and one ball per card (Theorem 2). As a consequence, SWISH remains NP-complete in these settings even when cards may contain more symbols. Together with previous results, this establishes a complete characterization of the computational complexity of SWISH with respect to both the number of symbols per card and the allowed symmetries.

The rest of the paper is organized as follows. In Section 2, we give definitions and formalize the problems. In Section 3, we state our results and prove them separately in subsections. In Section 4, we conclude the paper with remarks on future work.

## 2 Preliminaries

### 2.1 Swish

We formalize a generalization of SWISH in terms of the size of cards. Fix an  $h \times w$  integer grid, where  $(i, j)$  ( $1 \leq i \leq h$ ,  $1 \leq j \leq w$ ) denotes the point  $i$ -th from the top and  $j$ -th from the left;  $h = 4$  and  $w = 3$  in the original SWISH. As we mainly focus on the case where each card has exactly one hoop and exactly one ball, each *card*  $c$  is specified as an ordered pair of two points on the grid,  $((i_1, j_1), (i_2, j_2))$ , which means that  $c$  has a *hoop*  $\circ$  at  $(i_1, j_1)$  and a *ball*  $\bullet$  at  $(i_2, j_2)$ . If we are allowed to *rotate* a card  $c = ((i_1, j_1), (i_2, j_2))$ , we can use  $c$  as a card  $c^R := ((h - i_1 + 1, w - j_1 + 1), (h - i_2 + 1, w - j_2 + 1))$ ; here, it is restricted to 180-degree rotation even when  $h = w$ . If we are allowed to *flip*  $c$ , we can use  $c$  as a card  $c^F := ((i_1, w - j_1 + 1), (i_2, w - j_2 + 1))$ ; here, it is restricted to *horizontal* flip by symmetry (because if both horizontal and vertical flips are allowed, their combination results in the 180-degree rotation). If we are allowed both to rotate and to flip  $c$ , we can use  $c$  as, in addition to  $c^R$  and  $c^F$ , a card  $c^{RF} = ((h - i_1 + 1, j_1), (h - i_2 + 1, j_2))$ , which coincides with the card obtained by flipping  $c$  vertically.

Suppose that a set  $\mathcal{C}$  of cards is given. A subset  $\mathcal{S} \subseteq \mathcal{C}$  of cards is called a *swish* if the following conditions are satisfied by flipping or/and rotating some of the cards in  $\mathcal{S}$  if they are allowed: for every point  $p$  on the grid, either

- no card in  $\mathcal{S}$  has a hoop and no card in  $\mathcal{S}$  has a ball at  $p$ , or
- exactly one card in  $\mathcal{S}$  has a hoop and exactly one card in  $\mathcal{S}$  has a ball at  $p$ .

See Figure 1 again for examples.

The objective of the original game is to find a large swish. Then, we can consider natural decision problems as follows:

#### SWISH

**Input:** A set  $\mathcal{C}$  of cards on the  $h \times w$  grid and an integer  $k$ .

**Goal:** Decide whether there is a swish  $\mathcal{S} \subseteq \mathcal{C}$  such that  $|\mathcal{S}| \geq k$  or not, where both flipping and rotating any cards are allowed.

#### SWISH WITHOUT FLIP

**Input:** A set  $\mathcal{C}$  of cards on the  $h \times w$  grid and an integer  $k$ .

**Goal:** Decide whether there is a swish  $\mathcal{S} \subseteq \mathcal{C}$  such that  $|\mathcal{S}| \geq k$  or not, where rotating any cards is allowed but flipping them is not.

#### SWISH WITHOUT ROTATION

**Input:** A set  $\mathcal{C}$  of cards on the  $h \times w$  grid and an integer  $k$ .

**Goal:** Decide whether there is a swish  $\mathcal{S} \subseteq \mathcal{C}$  such that  $|\mathcal{S}| \geq k$  or not, where flipping any cards is allowed but rotating them is not.

#### SWISH WITHOUT FLIP OR ROTATION

**Input:** A set  $\mathcal{C}$  of cards on the  $h \times w$  grid and an integer  $k$ .

**Goal:** Decide whether there is a swish  $\mathcal{S} \subseteq \mathcal{C}$  such that  $|\mathcal{S}| \geq k$  or not, where neither flipping nor rotating any cards is allowed.

## 2.2 Graphs

We use the standard terms and notation on graphs. An undirected edge between two vertices  $u$  and  $v$  is denoted by an unordered pair  $\{u, v\}$ , and a directed edge from  $u$  to  $v$  is denoted by an ordered pair  $(u, v)$ .

In a graph (either undirected or directed), a *matching* is an edge subset such that each vertex has at most one incident edge in it. A matching is called *perfect* if all the vertices have exactly one incident edge. When each edge is assigned with a *weight* value, the *weight* of an edge subset is defined as the sum of the weights of edges therein.

In a directed graph, the *in-degree* of vertex  $v$  is the number of edges incoming to  $v$ , and the *out-degree* is the number of edges outgoing from  $v$ . A *directed cycle* is a connected directed graph in which every vertex has both in-degree and out-degree exactly 1. The *length* of a directed cycle is the number of vertices (or edges, equivalently) therein.

## 3 Algorithm and Complexity

The main results of this paper, solving an open problem in [3], are stated as follows:

► **Theorem 1.** *SWISH WITHOUT FLIP OR ROTATION is solved in polynomial time.*

► **Theorem 2.** *SWISH, SWISH WITHOUT FLIP, and SWISH WITHOUT ROTATION are NP-complete.*

Note that all the problems are clearly in NP, since if we are given a swish  $\mathcal{S}$  with additional information of how to use each card in  $\mathcal{S}$  (flipping or/and rotating it or not), we can easily confirm its validity in polynomial time.

### 3.1 Polynomial-Time Algorithm for Swish without Flip or Rotation

In this section, we prove Theorem 1.

Let  $(\mathcal{C}, k)$  be an instance of SWISH WITHOUT FLIP OR ROTATION, i.e.,  $\mathcal{C}$  is a set of cards and  $k$  is an integer. We construct from  $\mathcal{C}$  a bipartite graph  $G = (V^+, V^-; E)$  with edge weight  $w: E \rightarrow \mathbb{R}$  as follows (see Figure 2).

First, let  $V$  be the set of points on the grid. For each point  $p \in V$ , we create two copies  $p^+$  and  $p^-$ . Let  $V^+ := \{p^+ \mid p \in V\}$  and  $V^- := \{p^- \mid p \in V\}$ .

Next, for each card  $c = (p, q) \in \mathcal{C}$ , we create an edge  $e_c = \{p^+, q^-\}$  of weight  $w(e_c) = 1$ . Also, for each point  $p \in V$ , we create an edge  $f_p = \{p^+, p^-\}$  of weight  $w(f_p) = 0$ . Let  $E$  be the set of created edges.

Since it is well-known that a maximum-weight perfect matching in a bipartite graph can be computed in polynomial time [8], the following completes the proof.

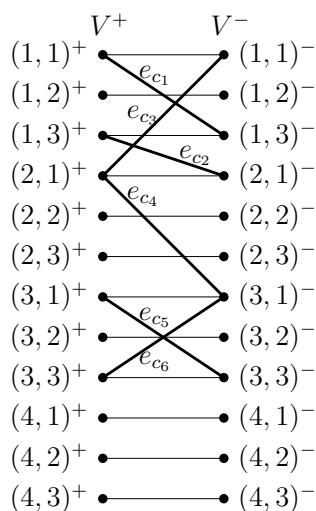
▷ **Claim 3.** There is a swish  $\mathcal{S} \subseteq \mathcal{C}$  such that  $|\mathcal{S}| = k$  without flipping or rotating any cards if and only if  $(G, w)$  has a perfect matching of weight  $k$ .

*Proof.* Roughly speaking, a swish of size  $k$  corresponds to a matching of weight  $k$  that can be extended to a perfect matching just by adding edges  $f_p$  of weight 0, and vice versa.

Let  $\mathcal{S} \subseteq \mathcal{C}$  be a swish such that  $|\mathcal{S}| = k$  without flipping or rotating any cards. We construct a perfect matching  $M \subseteq E$  in  $G$  with  $w(M) = k$ . For each card  $c = (p, q) \in \mathcal{S}$ , we add the edge  $e_c = \{p^+, q^-\}$  of weight 1 to  $M$ . Then, as  $\mathcal{S}$  is a swish without flip or rotation, for each point  $p \in V$ , either

- both  $p^+$  and  $p^-$  have no incident edge of the current  $M$ , or
- both  $p^+$  and  $p^-$  have exactly one incident edge of the current  $M$ .

Thus, we can make  $M$  a perfect matching by adding the edge  $f_p$  of weight 0 for each point  $p$  in the former situation. The weight of the resulting perfect matching is clearly  $|\mathcal{S}| = k$ .



■ **Figure 2** Bipartite graph  $G = (V^+, V^-; E)$  constructed from the Swish instance in Figure 1.  $\{e_{c_1}, e_{c_2}, e_{c_3}\} \cup \{f_p = \{p^+, p^-\} \mid p \in V \setminus \{(1,1), (1,3), (2,1)\}\}$  is a perfect matching, which corresponds to swish  $\{c_1, c_2, c_3\}$  without rotation or flip. The weight of this perfect matching is 3, which is equal to the size of the corresponding swish.

Conversely, let  $M \subseteq E$  be a perfect matching in  $G$  with  $w(M) = k$ . We construct a swish  $\mathcal{S} \subseteq \mathcal{C}$  such that  $|\mathcal{S}| = k$  without flipping or rotating any cards. Let  $\mathcal{S} := \{c \in \mathcal{C} \mid e_c \in M\}$ . By definition of edge weight  $w$ , we have  $|\mathcal{S}| = w(M) = k$ . Since  $M$  is a perfect matching and each edge  $f_p \in M$  connects two vertices  $p^+$  and  $p^-$  from the same point  $p \in V$ , if one of  $p^+$  and  $p^-$  is covered by some edge  $e_c \in M$ , then the other is also covered by some edge  $e_{c'} \in M$ . This implies that for each point  $p \in V$ , either

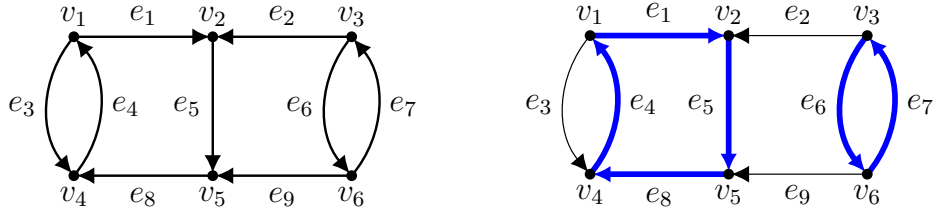
- no card in  $\mathcal{S}$  has a hoop and no card in  $\mathcal{S}$  has a ball at  $p$ , or
- exactly one card in  $\mathcal{S}$  has a hoop and exactly one card in  $\mathcal{S}$  has a ball at  $p$ .

Thus,  $\mathcal{S}$  is a swish without flip or rotation, which completes the proof.  $\triangleleft$

► **Remark 4.** The computational time is bounded by  $O(n^3)$  or  $O((m+n)n \log n)$  [5, 8], where  $n := |V| = hw$  and  $m$  is the number of input cards. Here, we may assume that  $h$  and  $w$  are bounded by  $O(m)$ , since a row or column that contains neither hoop nor ball can be removed without loss of generality. Thus, this is polynomial in the input size.

► **Remark 5.** We can consider more general settings of SWISH WITHOUT FLIP OR ROTATION. Suppose that we want to find a swish of size exactly  $k$ . Then, the same reduction results in the so-called *exact matching* problem, in which we are required to find a perfect matching of weight *exactly*  $k$ . For this problem, a *randomized* polynomial-time algorithm [10] is well-known, but the existence of a *deterministic* polynomial-time algorithm has been a major open problem in theoretical computer science for more than 40 years since its appearance [11].

As another generalization, suppose that each input card may have either no hoop but exactly two balls at different points or no ball but exactly two hoops at different points. Also in this case, the reduction and the proof are almost the same (e.g., if a card  $c$  has two hoops at points  $p$  and  $q$ , then we create an edge  $e_c = \{p^+, q^+\}$  of weight  $w(e_c) = 1$ ), but the resulting graph is no longer bipartite. Even though, it is known that a maximum-weight perfect matching can be computed in polynomial time (deterministic) [4], and that an exact-weight perfect matching can be computed in polynomial time (randomized) [10]. In both cases, it is



■ **Figure 3** Example of an instance (left) and its solution (right) of  $\forall$ EVEN DICYCLE-FACTOR.

known that the decision problem can be solved in  $O(n^\omega)$  time (randomized) [7, 13, 14] and an extra  $n$  factor is required to find a solution itself in the latter (exact-weight) case, where  $\omega \leq 2.371339$  is the matrix multiplication exponent [1].

### 3.2 NP-hardness of Swish without Rotation

In this section, we prove that SWISH WITHOUT ROTATION is NP-hard. The proof of Theorem 2 will be completed based on this proof in the next section.

We reduce the following problem, which is known to be NP-complete. For a directed graph  $G = (V, E)$ , we call an edge subset  $F \subseteq E$  an *even dicycle-factor* if  $F$  covers all vertices of  $V$  and each connected component of  $F$  is a directed cycle of even length (see Figure 3).

$\forall$ EVEN DICYCLE-FACTOR  
**Input:** A directed graph  $G = (V, E)$ .  
**Goal:** Decide whether there is an even dicycle-factor  $F \subseteq E$  or not.

► **Theorem 6** ([2, 6]).  $\forall$ EVEN DICYCLE-FACTOR is NP-complete.

Let  $G = (V, E)$  be an instance of  $\forall$ EVEN DICYCLE-FACTOR, i.e., a directed graph. We construct an equivalent instance  $(\mathcal{C}, k)$  of SWISH WITHOUT ROTATION as follows.

Let  $h := |V|$  and  $w := 4$ . We index the rows by  $V$  and the columns by  $\{1, 2, 3, 4\}$ , and simply denote by  $v^j$  each point  $(v, j)$  on the grid.

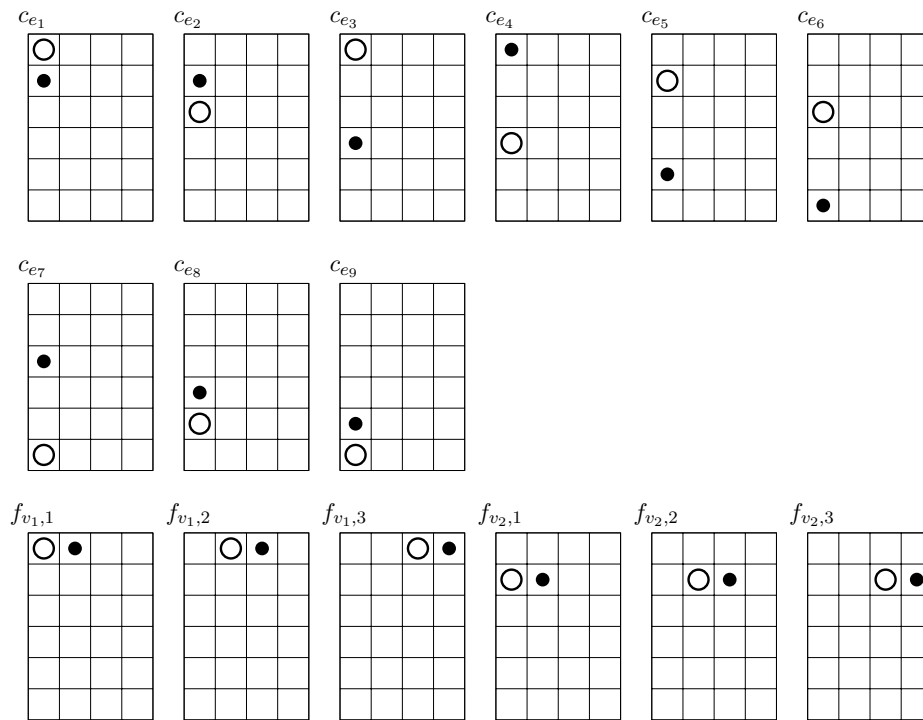
For each edge  $e = (u, v) \in E$ , we create a card  $c_e = (u^1, v^1)$ . In addition, for each vertex  $v \in V$ , we create three cards  $f_{v,j} = (v^j, v^{j+1})$  ( $j = 1, 2, 3$ ). Let  $\mathcal{C}$  be the set of created cards (see Figure 4).

The following claim completes the reduction, where we set  $k = 4|V|$  (see Figure 5).

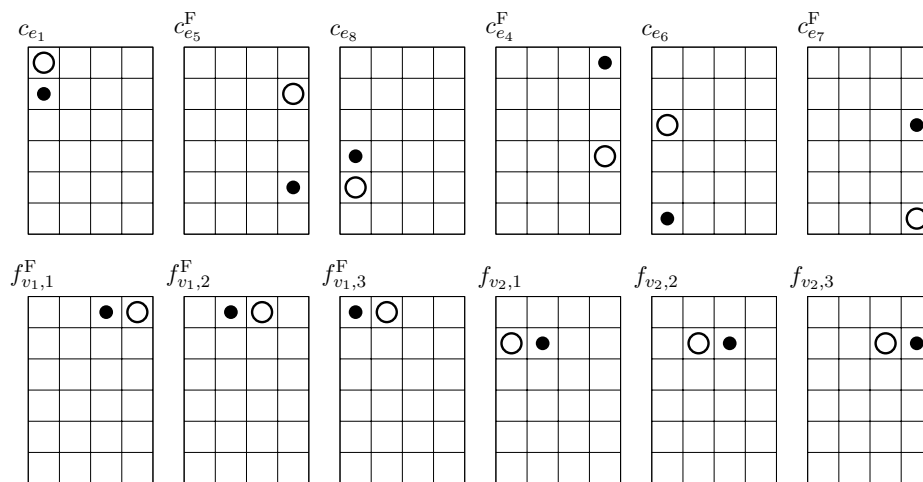
▷ **Claim 7.**  $G$  has an even dicycle-factor if and only if  $\mathcal{C}$  admits a swish of size  $4|V|$ , where flipping cards is allowed but rotating cards is forbidden.

*Proof.* Let  $F \subseteq E$  be an even dicycle-factor of  $G$ . Since each connected component of  $F$  is a directed cycle of even length, we can color the edges of  $F$  using two colors, say *red* and *blue*, such that any pair of adjacent edges receives different colors. Fix such a coloring, and let  $\mathcal{S} := \{c_e \mid e \in F\} \cup \{f_{v,j} \mid v \in V, j \in \{1, 2, 3\}\}$ . Note that  $|\mathcal{S}| = |F| + 3|V| = 4|V|$ . We prove that  $\mathcal{S}$  is a swish.

First, we fix how to use the card  $c_e = (u^1, v^1) \in \mathcal{S}$  for each  $e = (u, v) \in F$ . If  $e$  is red, we use it as it is, i.e., as  $c_e = (u^1, v^1)$ . In contrast, if  $e$  is blue, we use it by flipping, i.e., as  $c_e^F = (u^4, v^4)$ . As  $F$  is an (even) dicycle-factor, each vertex has exactly one incoming edge of  $F$  and exactly one outgoing edge of  $F$ , which are colored differently. Thus, following the above definition, for each vertex  $v \in V$ , one of the two points  $v^1$  and  $v^4$  receives exactly one



■ **Figure 4** The set of cards created from the instance of  $\forall$ EVEN DICYCLE-FACTOR in Figure 3, where we omit  $f_{v_i,j}$  for  $i = 3, 4, 5, 6$  (they are analogous to  $f_{v_1,j}$  and  $f_{v_2,j}$ ).



■ **Figure 5** The swish without rotation corresponding to the solution in Figure 3, where we omit  $f_{v_i,j}$  for  $i = 3, 4, 5, 6$ .

hoop and no ball and the other receives exactly one ball and no hoop. If  $v^1$  receives a ball (and  $v^4$  receives a hoop), then we use  $f_{v,j} = (v^j, v^{j+1}) \in \mathcal{S}$  ( $j = 1, 2, 3$ ) as they are, and otherwise, we use them by flipping, i.e., as  $f_{v,j}^F = (v^{5-j}, v^{4-j})$ . In both cases, the two middle points  $v^2$  and  $v^3$  receive exactly one hoop and exactly one ball from these three cards, and  $v^1$  and  $v^4$  receive exactly one of a hoop and a ball that is consistent with what is received from the cards  $c_e \in \mathcal{S}$ . Thus,  $\mathcal{S}$  is indeed a swish without rotation.

We now prove the reverse direction. Let  $\mathcal{S}$  be a swish of size  $4|V|$  without rotation, which means that we can put exactly one hoop and exactly one ball at every point on the grid only using flip. First, the middle points  $v^2$  and  $v^3$  for each vertex  $v \in V$  can be covered only by the three cards  $f_{v,j}$  ( $j = 1, 2, 3$ ), so we must use them. In particular, due to the constraint on the middle points, we can use the three cards either as they are or by flipping all of them. Then, we use exactly  $|V|$  cards in the form of  $c_e = (u^1, v^1)$  for edges  $e = (u, v) \in E$  in total such that for each vertex  $v \in V$ , one of  $v^1$  and  $v^4$  receives exactly one hoop and no ball and the other receives exactly one ball and no hoop.

Let  $F := \{e \in E \mid c_e \in \mathcal{S}\}$ . By the above argument, for each vertex  $v \in V$ , exactly one incoming edge (corresponding to a ball) and exactly one outgoing edge (corresponding to a hoop) is contained in  $F$ ; this means that every connected component of  $F$  is a directed cycle, which covers all the vertices. In addition, one of the two corresponding cards must be used as it is and the other must be flipped. According to this, we can color each directed cycle in  $F$  using two colors (flipped or not) such that any pair of adjacent edges receives different colors, which implies that the length must be even. Thus,  $F$  is an even dicycle-factor, which completes the proof.  $\triangleleft$

► **Remark 8.** It was not explicitly written in [2, 6] but the proofs of Theorem 6 imply that  $\forall$ EVEN DICYCLE-FACTOR is NP-hard even when the input graph is restricted so that the following conditions are both satisfied:

- every vertex has in-degree and out-degree at most 3 (by [6] immediately and also by [2] with preprocessing the input graph before the reduction, for the 2-linkage problem), and
- it is planar (by [6] combined with [9]).

Furthermore, it is easy to see that we can strengthen the first restriction on the input graph  $G = (V, E)$  as follows:

- $G$  is *tripartite*, where  $G$  admits a partition  $(V^-, V^\circ, V^+)$  of the vertex set  $V$  and every edge in  $E$  is either from  $V^-$  to  $V^\circ$ , from  $V^\circ$  to  $V^+$ , or from  $V^+$  to  $V^-$ ,
- every vertex in  $V^-$  has out-degree 1 and in-degree at most 3,
- every vertex in  $V^\circ$  has both in-degree and out-degree 1, and
- every vertex in  $V^+$  has in-degree 1 and out-degree at most 3.

This can be observed via a standard transformation of directed graphs: split each vertex  $v$  into three copies  $v^-$  (for incoming edges),  $v^\circ$  (for preserving the parity of length), and  $v^+$  (for outgoing edges) with two extra edges  $(v^-, v^\circ)$  and  $(v^\circ, v^+)$ , and replace each original edge  $(u, v)$  with  $(u^+, v^-)$ . This transformation does not change the problem.

Based on this restriction on the input graph  $G = (V, E)$  of  $\forall$ EVEN DICYCLE-FACTOR, we can impose the following restriction on the input set  $\mathcal{C}$  of cards of SWISH WITHOUT ROTATION: (\*) for each point  $p$  on the grid, either

- exactly one card has a hoop and at most two cards have a ball at  $p$ , or
- exactly one card has a ball and at most two cards have a hoop at  $p$ .

This can be seen by constructing an equivalent set  $\mathcal{C}'$  of cards from the set  $\mathcal{C}$  constructed in the proof.

By the tripartition and degree conditions, the edges from  $V^-$  to  $V^\circ$  and those from  $V^\circ$  to  $V^+$  form matchings. For such edges  $e = (v^-, v^\circ)$  and  $e' = (v^\circ, v^+)$  ( $v^- \in V^-$ ,  $v^\circ \in V^\circ$ ,  $v^+ \in V^+$ ), we include the cards  $c_e = ((v^-)^1, (v^\circ)^1)$  and  $c_{e'} = ((v^\circ)^1, (v^+)^1)$  in  $\mathcal{C}'$  as

in  $\mathcal{C}$ . In addition, we may assume that there exists a perfect matching between  $V^+$  and  $V^-$  consisting of edges from  $V^+$  to  $V^-$ , because otherwise  $G$  is clearly a no-instance of  $\forall\text{EVEN DICYCLE-FACTOR}$  ( $G$  does not admit any cycle-factors even without parity constraint), which can be tested in polynomial time. Fix any such perfect matching  $M$ , and for each edge  $e = (u^+, v^-) \in M$ , we include the card  $c_e = ((u^+)^1, (v^-)^1)$  in  $\mathcal{C}'$  as in  $\mathcal{C}$ . For each of the remaining edges  $e = (u^+, v^-) \in E \setminus M$  from  $V^+$  to  $V^-$ , we include the flipped card  $c_e^F = ((u^+)^4, (v^-)^4)$  in  $\mathcal{C}'$  instead of the card  $c_e \in \mathcal{C}$ . By construction, at this point,  $\mathcal{C}'$  satisfies the following:

- for each vertex  $v \in V = V^- \cup V^\circ \cup V^+$ , exactly one card has a hoop and exactly one card has a ball at  $v^1$ ;
- for each vertex  $v^- \in V^-$ , at most two cards have a ball and no card has a hoop at  $(v^-)^4$ ;
- for each vertex  $v^\circ \in V^\circ$ , any card has neither a hoop nor a ball at  $(v^\circ)^4$ ;
- for each vertex  $v^+ \in V^+$ , at most two cards have a hoop and no card has a ball at  $(v^+)^4$ .

In other words, the desired condition  $(*)$  will be satisfied if each  $v^1$  receives at most one of a hoop and a ball, each  $(v^-)^4$  receives exactly one hoop, each  $(v^\circ)^4$  receives exactly one of a hoop and a ball, and each  $(v^+)^4$  receives exactly one ball, respectively, in addition. It can be indeed achieved as follows: for each  $v \in V^- \cup V^\circ$ , we include the flipped cards  $f_{v,j}^F = (v^{5-j}, v^{4-j})$  ( $j = 1, 2, 3$ ) in  $\mathcal{C}'$  instead of  $f_{v,j} \in \mathcal{C}$ , and for each  $v^+ \in V^+$ , we include the cards  $f_{v^+,j} = ((v^+)^j, (v^+)^{j+1})$  ( $j = 1, 2, 3$ ) in  $\mathcal{C}'$  as in  $\mathcal{C}$ . Note that each of the middle points  $v^2, v^3$  receives exactly one hoop and exactly one ball in either case. Thus, we can construct a set  $\mathcal{C}'$  of cards that is equivalent to  $\mathcal{C}$  and satisfies the condition  $(*)$ .

### 3.3 NP-hardness of Swish

In this section, we prove that SWISH and SWISH WITHOUT FLIP are NP-hard as well as SWISH WITHOUT ROTATION shown in the previous section. The proof is almost the same, and the only difference is the coordination of points on the grid.

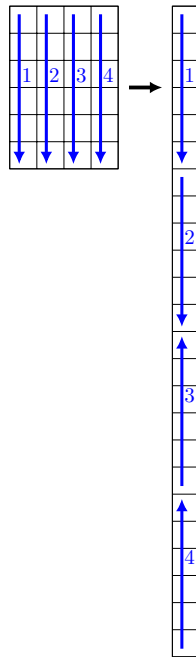
In the previous proof, we used the  $|V| \times 4$  grid, where each point  $(v, j)$  ( $v \in V, 1 \leq j \leq 4$ ) is denoted simply by  $v^j$ . Here we consider arranging these points on a single vertical line. Specifically, let  $h := 4|V|$  and  $w := 1$ , and name the  $4|V|$  points on the  $h \times w$  grid using  $v^j$  ( $v \in V, 1 \leq j \leq 4$ ) so that the point  $(i, 1)$  is named  $v^j$  if and only if the point  $(h - i + 1, 1)$  is named  $v^{5-j}$  (see Figures 6 and 7 for a concrete example). Then, the horizontal flip becomes meaningless, and the rotation plays the same role as the horizontal flip in the previous section. Thus, the same reduction works for this setting (rotation is allowed and horizontal flip is either allowed or not), which completes the proof of Theorem 2.

## 4 Concluding Remarks

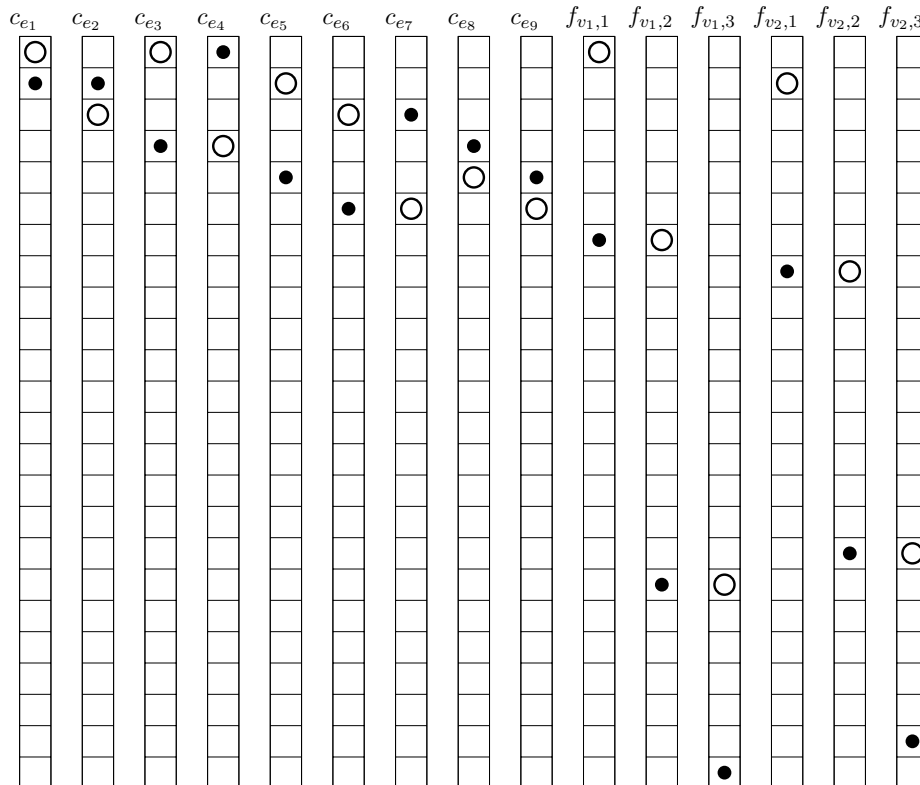
In this paper, we have completed the complexity classification of SWISH by resolving the previously open case of two symbols per card. Our results show that the computational complexity of SWISH is under interplay between the number of symbols per card and the geometric transformations allowed on cards. In particular, we have identified a sharp boundary: when neither flip nor rotation is allowed, SWISH with two symbols per card can be solved in polynomial time, whereas allowing even a single type of flip or 180-degree rotation already makes the problem NP-complete. Together with earlier results, this establishes a complete picture of the complexity of SWISH across all natural variants of the game.

There are two further questions on this line. One is whether the complexity changes if we allow overlap of two or more hoops and balls at the same point as long as the numbers of them are exactly the same (i.e., each hoop has exactly one ball as a partner, and vice versa);

25:10 Computational Complexity of Swish Is Solved



■ **Figure 6** An example of rearrangement of the points  $v^j$  ( $j = 1, 2, 3, 4$ ).



■ **Figure 7** The result of the rearrangement illustrated in Figure 6 for the cards in Figure 4.

the rule of the original game SWISH can also be interpreted in this way. For this question, the answer is No; that is, if neither flip nor rotation is allowed, the problem is in P, and otherwise it is NP-complete. This can be shown by similar reductions with appropriate modifications, which we will complete in the journal version. The other is the complexity of finding a swish (allowing flip and/or rotation) of any size or of size at least (or at most)  $k$  for a small integer  $k$ , e.g.,  $k = 3$ . While it is easy to find a swish of size at most  $k$  (if any) in  $O((4m)^k)$  time by brute-force search, it is open whether there exists an FPT (fixed-parameter tractable) algorithm or not. The other two types are basically open, but we will also remark partial observations in the journal version.

Beyond decision complexity, an intriguing line of research concerns the structure of card sets that admit no swish. Dailly, Lafourcade, and Marcadet [3] initiated a systematic study of such *no-swish* configurations. In their generalized setting, they constructed large families of card sets containing no swish, achieving asymptotically high densities. Moreover, for the original commercial deck of 60 cards, they identified a no-swish subset of size 28.

A natural and still open question is whether this value 28 is optimal. Is it possible to find a larger no-swish subset within the original SWISH deck, or does every set of 29 cards necessarily admit a swish? More broadly, determining tight bounds on the size of no-swish sets for fixed decks or restricted symmetry models appears to be a promising direction for future work, which would deepen our understanding of the combinatorial structure underlying the game and could require techniques different from those used in the generalized setting.

---

## References

- 1 Josh Alman, Ran Duan, Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. More asymmetry yields faster matrix multiplication. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2005–2039, 2025. doi:10.1137/1.9781611978322.63.
- 2 Jørgen Bang-Jensen and Stéphane Bessy. (Arc-)disjoint flows in networks. *Theoretical Computer Science*, 526:28–40, 2014. doi:10.1016/J.TCS.2014.01.011.
- 3 Antoine Dailly, Pascal Lafourcade, and Gael Marcadet. How did they design this game? Swish: complexity and unplayable positions. In *12th International Conference on Fun with Algorithms (FUN 2024)*, volume 291, pages 10:1–10:19, 2024. doi:10.4230/LIPIcs.FUN.2024.10.
- 4 Jack Edmonds. Maximum matching and a polyhedron with 0, 1-vertices. *Journal of Research of the National Bureau of Standards B*, 69(125-130):55–56, 1965.
- 5 Jack Edmonds and Richard M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 19(2):248–264, 1972. doi:10.1145/321694.321699.
- 6 Florian Hörsch, Csaba Király, Mirabel Mendoza-Cadena, Gyula Pap, Eszter Szabó, and Yutaro Yamaguchi. Odd and even harder problems on cycle-factors. *arXiv preprint arXiv:2510.18393*, 2025.
- 7 Chien-Chung Huang and Telikepalli Kavitha. Efficient algorithms for maximum weight matchings in general graphs with small edge weights. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1400–1412. SIAM, 2012. doi:10.1137/1.9781611973099.110.
- 8 Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.
- 9 Dániel Marx. NP-completeness of list coloring and precoloring extension on the edges of planar graphs. *Journal of Graph Theory*, 49(4):313–324, 2005. doi:10.1002/JGT.20085.
- 10 Ketan Mulmuley, Umesh V Vazirani, and Vijay V Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987. doi:10.1007/BF02579206.

## 25:12 Computational Complexity of Swish Is Solved

- 11 Christos H Papadimitriou and Mihalis Yannakakis. The complexity of restricted spanning tree problems. *Journal of the ACM*, 29(2):285–309, 1982. doi:10.1145/322307.322309.
- 12 Dana Rowland. Mathematical explorations with Swish. In *MAA MathFest (Mathematical Association of America)*, 2017.
- 13 Piotr Sankowski. Maximum weight bipartite matching in matrix multiplication time. *Theoretical Computer Science*, 410(44):4480–4488, 2009. doi:10.1016/J.TCS.2009.07.028.
- 14 Ryotaro Sato and Yutaro Yamaguchi. Exact matching in matrix multiplication time. *arXiv preprint arXiv:2508.04081*, 2025. doi:10.48550/arXiv.2508.04081.
- 15 Gali Shimoni and Zvi Shalem. Swish. Board game by ThinkFun. <https://www.thinkfun.com/products/swish/> (no longer available as of January 12, 2026), Rulebook: <https://legacy.thinkfun.com/wp-content/uploads/2015/09/Swish-1512-IN03.pdf>, Video: <https://www.youtube.com/watch?v=Hzi28pbeWWw> (available as of January 12, 2026), 2011.